

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ, МОЛОДІ ТА СПОРТУ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ

**МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
РОЗРАУНКОВО-ГРАФІЧНОЇ РОБОТИ**

Київ 2012

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ, МОЛОДІ ТА СПОРТУ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ РОЗРАУНКОВО-ГРАФІЧНОЇ РОБОТИ

для студентів напрямку підготовки 6.050803
«Акустотехніка» всіх форм навчання

Затверджено Вченою радою ФЕЛ НТУУ «КПІ»

Київ 2012

Основи мікропроцесорної техніки. Методичні вказівки до виконання розрахунково-графічної роботи для студентів напряму підготовки 6.050803 «Акусто-техніка» всіх форм навчання. - К.: НТУУ «КПІ», 2012. – 69 с.

Навчальне видання

Основи мікропроцесорної техніки. Методичні вказівки до виконання розрахунково-графічної роботи для студентів напряму підготовки 6.050803 «Акусто-техніка» всіх форм навчання

Укладачі: Терещенко Тетяна Олександрівна, проф., д. т. н.
Тодоренко Віктор Агафонович, доц., к.т.н.
Батрак Лариса Миколаївна, ст. викл.

Відповідальний редактор: Жуйков В.Я., проф., д. т. н.

Рецензент: Бондаренко В.М.

ЗМІСТ

ВСТУП	4
1. ЗАВДАННЯ НА РОЗРАХУНКОВО-ГРАФІЧНУ РОБОТУ	5
2. РОЗРОБКА СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ	22
2.1 Загальні рекомендації по створенню принципової схеми	22
2.2 Живлення елементів схем	22
2.3 Система скидання мікроконтролера	24
2.4 Генератор мікроконтролера	28
2.5 Приклади електричної принципової схеми контролера	30
2.5.1 Використання зовнішньої пам'яті програм	30
2.5.2 Використання зовнішньої пам'яті даних	32
2.5.3 Використання зовнішніх регістрів вводу/виводу даних	33
3. ОБРОБКА ІНФОРМАЦІЇ	34
4. ПРОГРАМУВАННЯ ПАРАЛЕЛЬНИХ ПОРТІВ	39
4.1 Ініціалізація портів	40
4.2 Робота з портами	42
5. ПРОГРАМУВАННЯ ТАЙМЕРІВ-ЛІЧИЛЬНИКІВ	44
6. ПРОГРАМУВАННЯ ПОСЛІДОВНОГО ПОРТУ	54
7. ПРОГРАМУВАННЯ СИСТЕМИ ПЕРЕРИВАНЬ	64
ЛІТЕРАТУРА	68

ВСТУП

Метою даної розрахунково-графічної роботи є закріплення і розширення знань, отриманих на лекціях, лабораторних і практичних заняттях, основних принципів побудови та функціонування мікропроцесорних пристроїв управління та обробки інформації та використання набутих практичних навиків побудови апаратної частини та програмного забезпечення мікропроцесорних систем. Крім того виконання даної роботи передбачає розвиток навиків самостійного аналізу роботи мікропроцесорних систем для передачі, обробки та зберігання інформації та написання елементарних програм мовою асемблера.

У даних методичних вказівках наведено варіанти завдань, основні вимоги до виконання роботи, а також теоретичні відомості, які необхідні для побудови апаратної частини пристрою і розробки програмного забезпечення. Теоретичні відомості надані для базового мікроконтролера AT89C51 сімейства MCS-51. Наведено рекомендації та приклади щодо розробки прикладного програмного забезпечення.

Виконання завдання передбачає звернення, як до лекційного матеріалу, так і самостійну роботу з науково-технічною літературою, довідниками та державними стандартами.

Контроль виконання, а також консультації здійснюються на спеціальних заняттях, що включаються у загальний розклад.

1. ЗАВДАННЯ НА РОЗРАХУНКОВО-ГРАФІЧНУ РОБОТУ

Завдання №1

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 2К, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (200/X - 1) \vee Y$, де: X – надходить на порт P1; Y – порт P2; Z – виводиться на порт P3
3	Ініціалізувати порт P0 на наступний режим роботи: <ul style="list-style-type: none">• Біти 0,1,2,3 – приймачі• Біти 4,5 – передавачі, на виході «0»• Біти 6,7 – передавачі, на виході «1»
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none">• Робота в якості таймера в «0» режимі роботи;• Константа завантаження -10fh;• Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none">• «1» режим роботи;• Швидкість обміну 9600 біт/с;• Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none">• Дозволити переривання від INT0 з вищим рівнем пріоритету;• Дозволити переривання від TC1 з нижчим рівнем пріоритету.

Завдання №2

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кристалічним резонатором, схемою скидання, зовнішньою пам'яттю даних 2К, внутрішньою пам'яттю програм.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (2 \cdot X + 1) \wedge Y$, де: X – надходить на порт P3; Y – на порт P1; Z – виводиться на порт P2
3	Ініціалізувати порт P1 на наступний режим роботи: <ul style="list-style-type: none">• Біти 0,1 – передавачі, на виході «0»;• Біти 2,3 – передавачі, на виході «1»;• Біти 6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none">• Робота в якості таймера в «1» режимі роботи;• Константа завантаження -110fh;• Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none">• «3» режим роботи;• Швидкість обміну 1200 біт/с;• Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none">• Дозволити переривання від INT1 з нижчим рівнем пріоритету;• Дозволити переривання від TC0 з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №3

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (2 \oplus X + 20) / Y$, де: X – надходить на порт P1; Y – на порт P2; Z – виводиться на порт P3.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1,2,3 – альтернативний режим роботи • Біти 4,5 – передавачі, на виході «0» • Біти 6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера у «2» режимі роботи; • Константа завантаження -0fh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «0» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від UART з вищим рівнем пріоритету; • Дозволити переривання від INT0 з нижчим рівнем пріоритету.

Завдання №4

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 4К, зовнішньою пам'яттю даних 4К.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = ((100+X)/10 - 60) \wedge Y$, де: X – надходить на порт P0; Y – на порт P2; Z – виводиться на порт P1.
3	Ініціалізувати порт P2 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – передавачі, на виході «1»; • Біти 2,3 – передавачі, на виході «0»; • Біти 4,5,6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера у «3» режимі роботи; • Константи завантаження -11h и 0fh; • Заборонити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 600 біт/с; • Дозволені переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від INT0 з нижчим рівнем пріоритету (переривання по рівню); • Дозволити переривання від INT1 з вищим рівнем пріоритету (переривання по фронту).

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №5

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, зовнішнім генератором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (130 \oplus X \vee Y + 1) / 3$, де: X – надходить на порт P3; Y – на порт P0; Z – виводиться на порт P1.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – альтернативний режим роботи; • Біти 2 - 5 – передавачі, на виході «1»; • Біти 6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника у «2» режимі роботи; • Константа завантаження -0fh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 4800 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC0 з вищим рівнем пріоритету; • Дозволити переривання від TC1 з нижчим рівнем пріоритету.

Завдання №6

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 16К, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (22 \oplus X \vee Y \wedge 33 - 2) \cdot 11$, де: X – надходить на порт P0; Y – на порт P3; Z – виводиться на порт P1.
3	Ініціалізувати порт P1 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,3 – передавачі, на виході «1»; • Біти 1,2 – передавачі, на виході «0»; • Біти 4,5,6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «1» режимі роботи; • Константа завантаження -101h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 6000 біт/с; • Дозволені переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від Rx (UART) з нижчим рівнем пріоритету; • Дозволити переривання від Tx (UART) з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №7

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, зовнішнім генератором, схемою скидання, зовнішньою пам'яттю програм (8К), зовнішньою пам'яттю даних (1К).
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = ((200 + 2) \oplus 3 + 1) / X$, де: X – надходить на порт P0; Y – на порт P1; Z – виводиться на порт P3.
3	Ініціалізувати порт P0 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – передавачі, на виході «1»; • Біти 2 - 5 – передавачі, на виході «0»; • Біти 6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «0» режимі роботи; • Константа завантаження -01fh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «0» режим роботи; • Швидкість обміну 4800 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від INT0 з нижчим рівнем пріоритету; • Дозволити переривання від INT1 з вищим рівнем пріоритету.

Завдання №8

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 1К, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (100 - 1 \oplus X) / 3 \vee 17$, де: X – надходить на порт P2; Y – на порт P3; Z – виводиться на порт P1
3	Ініціалізувати порт P2 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 4,5 – передавачі, на виході «1»; • Біти 1,2 – передавачі, на виході «0»; • Біти 0,3,6,7 – приймачі.
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «3» режимі роботи; • Константа завантаження -101h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 1800 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від UART з нижчим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №9

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, зовнішньою пам'яттю даних (2K).
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (2 \vee 20 \wedge 30 \oplus X + 1) \cdot Y$, де: X – надходить на порт P2; Y – на порт P1; Z – виводиться на порт P0.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – альтернативна функція; • Біти 2 - 5 – приймачі; • Біти 6,7 – передавачі, на виході «0».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «1» режимі роботи з зовнішнім управлінням; • Константа завантаження -0B1fh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 2300 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC0 з вищим рівнем пріоритету.

Завдання №10

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, зовнішньою пам'яттю програм 16K, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (((2+39) / Y) + 1) \vee 33$, де: X – надходить на порт P0; Y – надходить на порт P3; Z – виводиться на порт P2
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 4,5 – передавачі, на виході «1»; • Біти 1,2 – приймачі; • Біти 0,3,6,7 – передавачі на виході «1».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера у «2» режимі роботи; • Константа завантаження -0A1h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 600 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від INT0 з нижчим рівнем пріоритету, переривання по рівню.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №11

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, внутрішньою пам'яттю програм та пам'яттю даних, зовнішнім регістром (передавачем), підключеним по шині P8.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (X \vee Y + 1) \cdot 2$, де: X – надходить на порт P1; Y – надходить на порт P2; Z – виводиться на порт P0.
3	Ініціалізувати порт P0 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 1 - 4 – приймачі; • Біти 5 - 7 – передавачі, на виході «0».
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «1» режимі роботи з зовнішнім управлінням; • Константа завантаження - 0B1fh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 1250 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC0 з нижчим рівнем пріоритету; • Дозволити переривання від TC1 з вищим рівнем пріоритету.

Завдання №12

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, зовнішнім генератором, схемою скидання, зовнішньою пам'яттю програм 64К, внутрішньою пам'яттю даних, зовнішнім регістром (приймачем), підключеним по шині P8.
2	Мовою асемблеру MCS51 написати програму, що реалізує обробку елементів двох масивів A[i], B[i], розміщених у внутрішній пам'яті програм. Результуючий масив Z[i] розмістити у внутрішній пам'яті даних. Математична операція $Z[i] = A[i] + B[i]$, де: i = 10.
3	Ініціалізувати порт P1 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – передавачі на виході «0»; • Біти 2,3 – приймачі; • Біти 4 - 7 – передавачі, на виході «1».
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «3» режимі роботи; • Константа завантаження - 0Ah; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 1600 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від INT1 з вищим рівнем пріоритету, переривання по фронту. • Дозволити переривання від UART з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №13

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 2К, внутрішньою пам'яттю даних (2К).
2	Мовою асемблеру MCS51 написати програму, що реалізує обробку елементів двох масивів A[i], B[i], розміщених у внутрішній пам'яті даних. Результуючий масив Z[i] розмістити в зовнішній пам'яті даних, починаючи з адреси 1AAAh. Математична операція $Z[i] = A[i] * B[i]$, де: $i = 4$.
3	Ініціалізувати порт P0 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – приймачі; • Біти 2 - 5 – передавачі, на виході «1»; • Біти 6,7 – передавачі, на виході «0».
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «1» режимі роботи з зовнішнім управлінням; • Константа завантаження -0h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «1» режим роботи; • Швидкість обміну 1200 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від UART з вищим рівнем пріоритету; • Дозволити переривання від INT1 з вищим рівнем пріоритету.

Завдання №14

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, зовнішньою пам'яттю програм 4К, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує обробку елементів масивів A[i], B[i], C[i] розміщених у внутрішній пам'яті даних. Результуючий масив Z[i] розмістити в зовнішній пам'яті даних, починаючи з адреси 1010h. Вираз $Z[i] = (A[i] + B[i]) * C[i]$, де: $i = 8$.
3	Ініціалізувати порт P2 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,7 – приймачі; • Біти 1,4,5 – передавачі, на виході «1»; • Біти 2,3,6 – передавачі, на виході «0»;
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «1» режимі роботи з програмним управлінням; • Константа завантаження -0A7h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «1» режим роботи; • Швидкість обміну 2320 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по фронту від INT1 з нижчим рівнем пріоритету

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №15

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм (8K), зовнішньою пам'яттю даних (4K).
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (2 + 2 \cdot 10 \oplus X - 2) \wedge Y$, де: X – надходить на порт P3; Y – на порт P2; Z – виводиться на порт P1.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 6,7 – альтернативна функція; • Біти 1 - 3 – приймачі; • Біти 0, 4, 5 – передавачі, на виході «0».
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «0» режимі роботи з зовнішнім управлінням; • Константа завантаження -11h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 9600 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC1 з нижчим рівнем пріоритету.

Завдання №16

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, внутрішнім генератором з кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних, зовнішнім регістром (приймачем) и зовнішнім регістром (передавачем), підключеними по шині P8.
2	Мовою асемблеру MCS51 написати програму, що реалізує обробку елементів масивів A[i], B[i], C[i] розміщених у зовнішній пам'яті програм (A[i], B[i],) та зовнішній пам'яті даних (C[i]). Адреси масивів вибрати довільно. Результуючий масив Z[i] розмістити у зовнішній пам'яті даних, починаючи з адреси 1010h. Вираз $Z[i] = A[i] * B[i] - C[i]$, де: i = 8.
3	Ініціалізувати порти P0, P2 на наступні режими роботи: <ul style="list-style-type: none"> • Порт P0 – передавачі, на виході «1»; • Порт P2 – приймачі.
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера у «3» режимі роботи; • Константа завантаження -0Ah; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 4600 біт/с; • «0» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від UART з нижчим рівнем пріоритету; • Дозволити переривання від TC0 з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №17

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 2К, зовнішньою пам'яттю даних (2К), двома зовнішніми регістрами (приймачами), підключеними по шині P8.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = X / (Y \sqrt{120 + 1})$, де: X – надходить на порт P1; Y – на порт P2; Z – виводиться на порт P3.
3	Ініціалізувати порти P0, P1, P2 на наступні режими роботи: <ul style="list-style-type: none"> • Порт P0 – приймачі; • Порт P1 – передавачі, на виході «1»; • Порт P2 – передавачі, на виході «0»;
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера у «2» режимі роботи з зовнішнім управлінням; • Константа завантаження -0AB1h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 12000 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по фронту від INT0 з вищим рівнем пріоритету; • Дозволити переривання по рівню від INT1 з нижчим рівнем пріоритету.

Завдання №18

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = 45 + 122 / Y + 4 \sqrt{13}$, де: X – надходить на порт P3; Y – надходить на порт P2; Z – виводиться на порт P1
3	Ініціалізувати порт P2 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – передавачі на виході «0»; • Біти 2,3 – приймачі; • Біти 4 - 7 – передавачі, на виході «1».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «3» режимі роботи з зовнішнім управлінням; • Константа завантаження -11h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «0» режим роботи; • Швидкість обміну 4000 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC0 з нижчим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №19

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, зовнішньою пам'яттю даних (64К).
2	Мовою асемблеру MCS51 написати програму, що обробляє елементи масивів A[i], B[i], C[i] розміщених у зовнішній пам'яті програм (адреси вибрати самостійно). Результуючий масив Z[i] розмістити у зовнішній пам'яті даних, починаючи з адреси 11h. Вираз $Z[i] = A[i] \vee B[i] * C[i]$ де: $i = 6$.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 2,3 – альтернативна функція; • Біти 0, 1 – приймачі; • Біти 4 - 7 – передавачі, на виході «0».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «1» режимі роботи з програмним управлінням; • Константа завантаження -0AAAAh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 13500 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC1 з вищим рівнем пріоритету • Дозволити переривання від UART з вищим рівнем пріоритету.

Завдання №20

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, зовнішньою пам'яттю програм 1К, внутрішньою пам'яттю даних. По шині P8 підключені 3 зовнішні регістри – приймачі.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = 2 + X / Y \vee 13$, де: X – надходить на порт P1; Y – надходить на порт P2; Z – виводиться на порт P0.
3	Ініціалізувати порти P0, P3 на наступний режим роботи: <ul style="list-style-type: none"> • Порт P0 – приймачі; • Порт P3 – режим альтернативних функцій.
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «0» режимі роботи; • Константа завантаження -0BAA1h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 22600 біт/с; • Дозволені переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по рівню від INT0 з нижчим рівнем пріоритету; • Дозволити переривання по фронту від INT1 з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №21

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, зовнішнім генератором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних, двома зовнішніми регістрами (передавачі), підключеними по шині P8
2	Мовою асемблеру MCS51 написати програму, що обробляє елементи масивів A[i], B[i], C[i] розміщених у зовнішній пам'яті даних (адреси вибрати самостійно). Результуючий масив Z[i] розмістити у внутрішній пам'яті даних, починаючи з адреси 23h. Вираз $Z[i] = A[i] \oplus B[i] / C[i]$, де: $i = 9$.
3	Ініціалізувати порти P0 - P3 на наступні режими роботи: <ul style="list-style-type: none"> • Порт P0 – приймачі; • Порт P1 – передавачі, на виході «1»; • Порт P2 – передавачі, на виході «0»; • Порт P3 – альтернативні функції.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «0» режимі роботи з зовнішнім управлінням; • Константа завантаження -0h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «1» режим роботи; • Швидкість обміну 12300 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC1 з вищим рівнем пріоритету.

Завдання №22

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, зовнішньою пам'яттю даних 8К.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = 1 + X / Y + 1 \vee 31$, де: X – надходить на порт P0; Y – на порт P3; Z – виводиться на порт P2.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – альтернативна функція; • Біти 2,3 – приймачі; • Біти 4 - 7 – передавачі, на виході «0»
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «1» режимі роботи з зовнішнім управлінням; • Константа завантаження -01h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 11600 біт/с; • «0» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від UART з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №23

1	Розробити принципову електричну схему контролера (АТ89С51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, зовнішньою пам'яттю даних (32К), двома зовнішніми регістрами (перший - передавач, другий - приймач), підключеними по шині P8.
2	Мовою асемблеру MCS51 написати програму, що обробляє елементи масивів A[i], B[i], C[i] розміщених у зовнішній пам'яті програм (адреси вибрати довільно). Результуючий масив Z[i] розмістити у зовнішній пам'яті даних, починаючи з адреси 223h. Вираз $Z[i] = A[i] / B[i] + C[i]$, де: $i = 16$.
3	Ініціалізувати порт P0 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – приймачі; • Біти 2 - 4 – передавачі, на виході «1»; • Біти 5, 6,7 – передавачі, на виході «0».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «0» режимі роботи з програмним керуванням; • Константа завантаження -0111h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 22300 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC1 з вищим рівнем пріоритету.

Завдання №24

1	Розробити принципову електричну схему контролера (АТ89С51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 16К, зовнішньою пам'яттю даних (2К).
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = ((X+10) \vee Y + 1) * 5$, де: X – надходить на порт P1; Y – надходить на порт P3; Z – виводиться на порт P0.
3	Ініціалізувати порт P1 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0-3 передавачі, на виході «0»; • Біти 4,5 – приймачі; • Біти 6,7 – передавачі, на виході «1»
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «3» режимі роботи з зовнішнім управлінням; • Константа завантаження -01h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 12000 біт/с; • «0» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по рівню від INT0 з вищим рівнем пріоритету; • Дозволити переривання по фронту від INT1 з вищим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №25

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 64К, зовнішньою пам'яттю даних (1К).
2	Мовою асемблеру MCS51 написати програму, що обробляє елементи масивів A[i], B[i], розміщених у зовнішній пам'яті програм і C[i], розміщеного у внутрішній пам'яті даних. Результуючий масив Z[i] розмістити у зовнішній пам'яті даних, починаючи з адреси 100h. Вираз $Z[i] = A[i] + B[i] - C[i]$, де $i = 6$
3	Ініціалізувати порти P0 - P3 на наступні режими роботи: <ul style="list-style-type: none"> • Порт P1 – приймачі; • Порт P0 – передавачі, на виході «1»; • Порт P2 – передавачі, на виході «0»; • Порт P3 – альтернативні функції.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «0» режимі роботи з зовнішнім управлінням; • Константа завантаження -0h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 6800 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по фронту від INT0 з вищим рівнем пріоритету.

Завдання №26

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм 16К, зовнішньою пам'яттю даних 8К, одним зовнішнім регістром – приймачем, підключеним по шині P8.
2	Мовою асемблеру MCS51 написати програму, яка обробляє елементи масивів A[i], розміщений у зовнішній пам'яті програм і B[i], C[i], розміщених у внутрішній пам'яті даних. Результуючий масив Z[i] розмістити у внутрішній пам'яті даних, починаючи з адреси 100h. Вираз $Z[i] = A[i] * B[i] / C[i]$, де $i = 10$.
3	Ініціалізувати порт P1 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0 - 3 – передавачі, на виході «1»; • Інші Біти – приймачі.
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «3» режимі роботи з програмним управлінням; • Константа завантаження -0h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 3600 біт/с; • «2» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по рівню від INT0 з нижчим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №27

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, зовнішнім генератором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних, трьома зовнішніми регістрами – передавачами, підключеними по шині P8 .
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = (120 - 20 \wedge 10 \oplus X + 2) \cdot Y$, де: X – надходить на порт P0; Y – надходить на порт P1; Z– виводиться на порт P3.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – приймачі; • Біти 2 - 4 – альтернативна функція; • Біти 5 - 7 – передавачі, на виході «1».
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «3» режимі роботи з зовнішнім управлінням; • Константа завантаження -02h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «0» режим роботи; • Швидкість обміну 1000 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від INT0, INT1 з вищим рівнем пріоритету.

Завдання №28

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю даних 16K, внутрішньою пам'яттю програм.
2	Мовою асемблеру MCS51 написати програму, що обробляє елементи масивів A[i], розміщеного у зовнішній пам'яті програм і B[i], C[i], розміщених у зовнішній пам'яті даних (адреси вибрати самостійно). Результуючий масив Z[i] розмістити у зовнішній пам'яті даних, починаючи з адреси 200h. Вираз $Z[i] = A[i] \oplus B[i] + C[i]$, де: i = 15.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біт 0 – передатчик на виході «1»; • Біт 1 – приймачі; • Біт 3 – передатчик на виході «1»
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «3» режимі роботи з програмним управлінням; • Константа завантаження -0AA1h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 24600 біт/с; • «2» режим роботи; • Дозволені переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по фронту від INT1 з нижчим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №29

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних, двома зовнішніми регістрами (приймачами), підключеними по шині P8.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = X/Y + 1 \sqrt{10} + 2 \cdot Y$, де: X – надходить на порт P0; Y – надходить на порт P1; Z – виводиться на порт P3.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біт 0 – альтернативна функція; • Біт 2 – приймач; • Біт 6 – передавач, на виході «0».
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «2» режимі роботи з зовнішнім управлінням; • Константа завантаження -0Bh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «0» режим роботи; • Швидкість обміну 10000 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC1 з вищим рівнем пріоритету; • Дозволити переривання по фронту від INT0 з нижчим рівнем пріоритету.

Завдання №30

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, зовнішньою пам'яттю програм 8K, внутрішньою пам'яттю даних.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = X + Y / (1 + 10) / 2 \sqrt{33}$, де: X – надходить на порт P1; Y – надходить на порт P2; Z – виводиться на порт P3.
3	Ініціалізувати порт P0 на наступний режим роботи: <ul style="list-style-type: none"> • Біт 4 – передавач, на виході «0»; • Біт 0 – приймач; • Біт 3 – передавач, на виході «1».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «0» режимі роботи з зовнішнім управлінням; • Константа завантаження -01h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 300 біт/с; • «1» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по рівню від INT0 з вищим рівнем пріоритету; • Дозволити переривання від TC0 з нижчим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №31

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, зовнішньою пам'яттю програм (2K), зовнішньою пам'яттю даних (2K).
2	Мовою асемблеру MCS51 написати програму, що обробляє елементи масивів A[i], B[i], C[i], розміщених у зовнішній пам'яті даних (адреси вибирати самостійно). Результуючий масив Z[i] розмістити у внутрішній пам'яті програм, починаючи з адреси 100h. Вираз $Z[i] = A[i] + B[i] * C[i]$, де: $i = 4$.
3	Ініціалізувати порт P1 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 2 - 5 – приймачі; • Біти 6,7 – передавачі, на виході «1».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості лічильника в «1» режимі роботи з програмним управлінням; • Константа завантаження -01h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «1» режим роботи; • Швидкість обміну 1000 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC1 з нижчим рівнем пріоритету.

Завдання №32

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних, одним зовнішнім регістром (передавачем), підключеним по шині P8 .
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = ((X + 9) / Y + 1) \vee 1$, де: X – надходить на порт P0; Y – на порт P3; Z – виводиться на порт P2
3	Ініціалізувати порти P0 - P3 на наступні режими роботи: <ul style="list-style-type: none"> • Порт P2 – приймачі; • Порт P1 – передавачі, на виході «1»; • Порт P0 – передавачі, на виході «0»; • Порт P3 – альтернативні функції.
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «0» режимі роботи з зовнішнім управлінням; • Константа завантаження -0h; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «2» режим роботи; • Швидкість обміну 1600 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання по рівню від INT0 з нижчим рівнем пріоритету; • Дозволити переривання по фронту від INT0 з нижчим рівнем пріоритету.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Завдання №33

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, кварцовим резонатором, схемою скидання, внутрішньою пам'яттю програм, внутрішньою пам'яттю даних, двома зовнішніми регістрами (1 приймач та 1 передавач), підключеними по шині P8.
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = 1 \vee 2 \wedge 3 \oplus X + 4 \cdot Y$, де: X – надходить на порт P0; Y – надходить на порт P1; Z – виводиться на порт P3.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,1 – альтернативна функція; • Біт 2 – приймач; • Біт 6 – передавач, на виході «1».
4	Ініціалізувати таймер/лічильник TC1 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «3» режимі роботи з зовнішнім управлінням; • Константа завантаження -0Bfh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • «3» режим роботи; • Швидкість обміну 32300 біт/с; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від TC0 з вищим рівнем пріоритету; • Дозволити переривання від TC1 з нижчим рівнем пріоритету.

Завдання №34

1	Розробити принципову електричну схему контролера (AT89C51) з колами живлення, керамічним резонатором, схемою скидання, зовнішньою пам'яттю програм 64K, зовнішньою пам'яттю даних (8K).
2	Мовою асемблеру MCS51 написати програму, що реалізує математичні операції $Z = 33 \vee X / (Y + 1)$, де: X – надходить на порт P1; Y – надходить на порт P3; Z – виводиться на порт P0.
3	Ініціалізувати порт P3 на наступний режим роботи: <ul style="list-style-type: none"> • Біти 0,5 – передавачі, на виході «1»; • Біти 1,6 – приймачі; • Інші біти – передавачі, на виході «0»
4	Ініціалізувати таймер/лічильник TC0 на наступний режим роботи: <ul style="list-style-type: none"> • Робота в якості таймера в «0» режимі роботи; • Константа завантаження -0FFAh; • Дозволити переривання.
5	Ініціалізувати UART контролер з наступним протоколом обміну: <ul style="list-style-type: none"> • Швидкість обміну 6600 біт/с; • «1» режим роботи; • Дозволити переривання.
6	Ініціалізувати систему переривань на наступний режим роботи: <ul style="list-style-type: none"> • Дозволити переривання від UART з нижчим рівнем пріоритету.

2. РОЗРОБКА СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ

2.1. Загальні рекомендації по створенню принципової схеми

Принципова електрична схема повинна мати закінчене функціональне призначення. На принциповій схемі необхідно показати:

- окремі функціональні вузли і блоки;
- виділити джерела живлення і напруги, які вони генерують;
- визначити призначення кожного елемента схеми;
- простежити, на які вузли подано напругу живлення;
- визначити входи і виходи системи;
- визначити сигнали, їх джерела та об'єкти керування.

У пояснювальній записці слід відзначити:

- які перетворення виконують окремі вузли принципової схеми і куди подається вихідний сигнал, де і чим реєструється.

Студент повинен знати призначення кожної мікросхеми, її характеристики, принцип роботи, вказати призначення кожного елемента в схемі, вплив кожного елемента на працездатність схеми. Привести опис технічних характеристик кожного елемента схеми.

2.2. Живлення елементів схеми

Для живлення елементів схеми контролера може використовуватися велика кількість каналів стабілізованої напруги. Серед цих каналів виділяється джерело живлення мікроконтролера та оточуючих його базових мікросхем. Для живлення мікроконтролерів сімейства MCS51 зазвичай використовують стабілізовану напругу +3,3 або +5В. Мікроконтролер AT89C51 фірми Atmel сімейства MCS51 використовує напругу живлення +5В.

Схемотехніка джерела живлення суттєво залежить від типу джерела первинного живлення.

На рис. 2.1, 2.2 зображено схеми джерел живлення де використовується мережа змінного струму 220В, 50Гц.

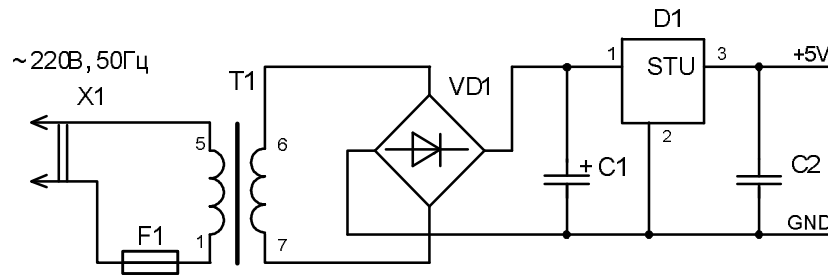


Рис.2.1

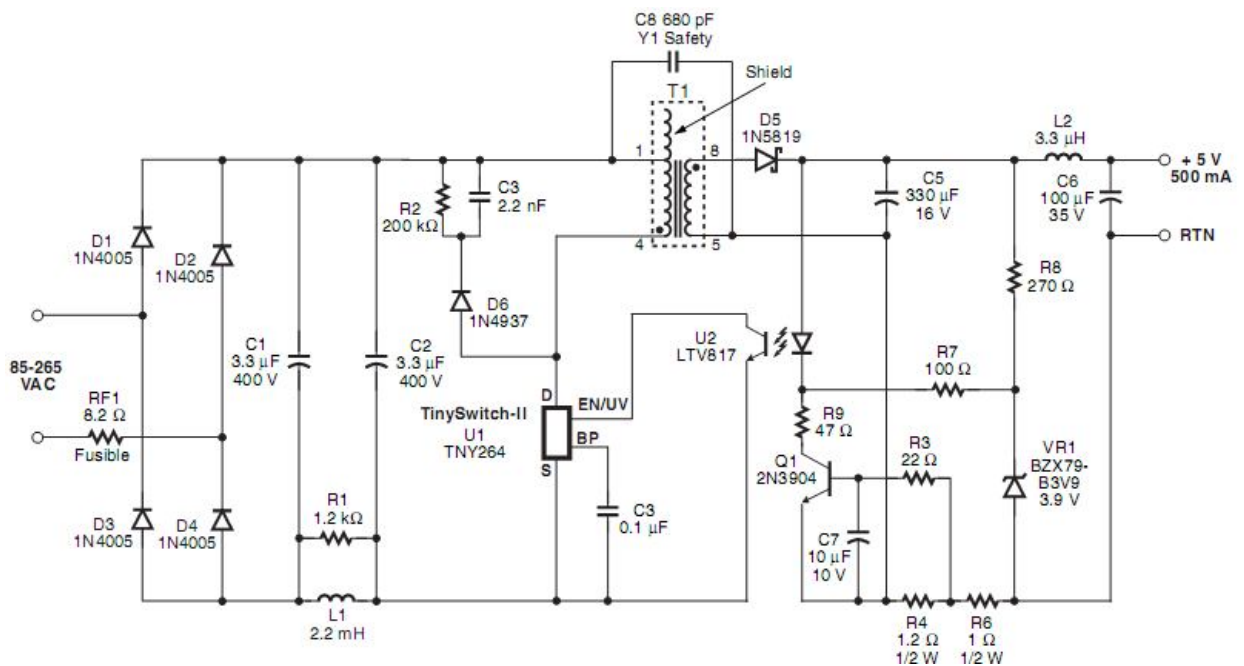


Рис.2.2

На рис.2.1 приведено схему де використовується лінійний стабілізатор напруги живлення, наприклад LM78L05. Цю схему доцільно використовувати якщо споживана потужність мікроконтролерної системи знаходиться у межах 0,5 - 3Вт.

На рис.2.2 зображено схему зворотньоходового одноканального стабілізатора напруги, побудованого з використанням спеціалізованої мікросхеми TNY264 фірми Power Integration. Цю схемотехніку доцільно використовувати якщо споживана потужність мікроконтролерної системи знаходиться у межах 2,5 - 50Вт. На сайті фірми Power Integration можна знайти дистрибутив спеці-

алізованої програми PI Expert v.8 призначеної для автоматизованого розрахунку схем такого типу.

На рис. 2.3, 2.4 зображено схеми джерел живлення де використовується мережа сталого струму.

На рис.2.3 приведено схему, де використовується лінійний стабілізатор напруги живлення. Ця схема входить до складу приведеної раніше на рис.2.1.

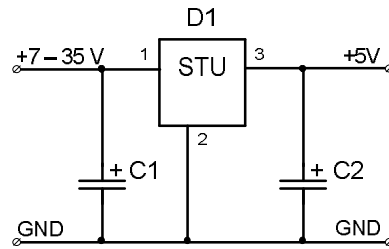


Рис.2.3

На рис.2.4 зображено схему імпульсного стабілізатора напруги, побудованого на базі спеціалізованої мікросхеми LM2574 фірми Motorola.

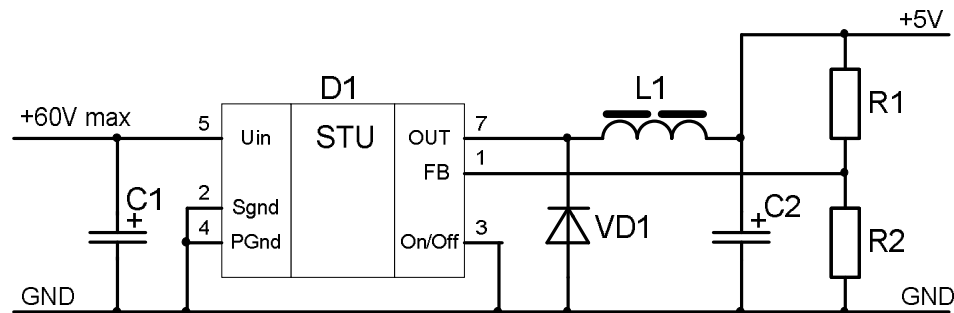


Рис.2.4

Ця мікросхема випускається з фіксованими рівнями вихідної напруги, та у версії з можливістю зовнішнього регулювання. Таку схемотехніку доцільно використовувати якщо споживана потужність мікроконтролерної системи знаходиться у межах 1 - 30Вт.

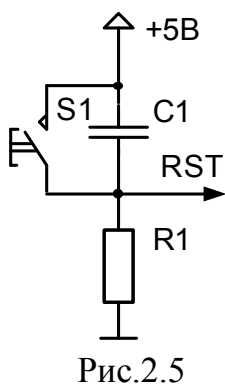
2.3. Система скидання мікроконтролера

Система скидання мікроконтролера у переважній більшості випадків повинна забезпечувати наступні функції:

- Автоматичне скидання при подачі напруги живлення;

- Можливість ручного скидання;
- Скидання у разі провалів напруги нижче заданого рівня;
- Формування імпульсів скидання заданої тривалості;
- Можливість скидання при зависанні мікроконтролера. внаслідок роботи сторожового таймера (WDT).

В простих схемах звичайно забезпечують лише можливість скидання при подачі напруги живлення. В мікроконтролерах сімейства MCS51, що виконуються по CMOS технології для цього застосовують послідовний RC-ланцюг. Схему такого ланцюга приведено на рис.2.5. Зображена схема забез-



печує також можливість ручного скидання мікроконтролера. Ця схема має суттєві недоліки. По-перше вона не забезпечує первинне скидання у разі повільного зростання напруги живлення мікроконтролера. По-друге, схема скидання при тимчасових провалах напруги живлення не забезпечує повторне скидання мікроконтролера.

Ці недоліки відсутні у разі застосування супервізорів напруги. Звичайно супервізори напруги контролюють напругу живлення. У разі, якщо напруга живлення досягає певного рівня починає формуватися інтервал часу затримки завершення імпульсу скидання. У простих схемах супервізорів цей інтервал жорстко обумовлений. У більш складних супервізорах існує можливість зовнішньої корекції. Супервізори випускають з можливістю генерації імпульсів скидання позитивних або негативних на різні значення порогової напруги.

Наприклад, супервізор ADM6825 фірми Analog Devices випускається з наступними значеннями порогових рівней напруги: 1,58В, 1,67В, 2,19В, 2,32В, 2,63В, 2,93В, 3,08В, 4,38В. Він також забезпечує тривалість інтервалу затримки 140мкс, можливість побудови схеми ручного скидання та може конфігуруватися для генерації як позитивних, так і негативних імпульсів ски-

дання мікроконтролерів. На рис.2.6 зображено схеми застосування цього супервізора при генеруванні як негативного, так і позитивного сигналів скидання.

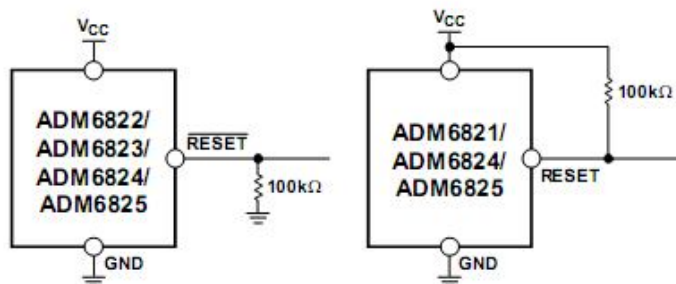


Рис.2.6

У табл.2.1 приведено параметри деяких сучасних супервізорів напруги.

Таблиця 2.1

Тип	V _{CC} , [В]	RESET		WDT	Тип корпусу
		Тривалість імпульсу скидання, [мс]	Рівень імпульсу низький/високий		
MAX69x*	4,4...4,6 5	50..200	Н/В	+	DIP-8. DIP-16 и др.
MAX814	3,3...5	програмована	Н/В	-	DIP-8
MAX825	5	програмована	Н/В	+	SOT23-5
TC1270	1,2...5,5	280	Н/-	-	SOT143-4
TC1271	1,2...5,5	280	-/В	-	SOT143-4
TCM811	1,2...5,5	280	Н/-	-	SOT123A-3
TC51	0,7...10	50	Н/-	-	TO92-3
TC32M	4,5...5,5	500	Н/-	+	TO92 DIP-8-3
TC1232	4,5...5,5	610	Н/В	+	DIP-8
TL77x*	3,6...18	програмована	Н/В	-	DIP-8,SOP-8 и др.
MCP810	1...5,5	350	-/В	-	SOT143-4
ADM1232	5	програмована	Н/В	+	DIP-8

У деяких випадках супервізор напруги об'єднують в одній інтегральній схемі з безперервним стабілізатором напруги. Прикладом такої мікросхеми може бути стабілізатор напруги з вбудованим супервізором 78MR05.

При роботі мікроконтролерів в умовах сильних імпульсних завад можливі збої в роботі генератора та системи синхронізації. Такі збої можуть привести до неправильного виконання програми мікроконтролером. Для вирішення цієї проблеми використовують сторожовий таймер – Watchdog Timer. Сторожовий таймер формує періодичну послідовність імпульсів скидання мікроконтролера з програмованим, або жорстко визначеним періодом повторення 1мс – 1хвилини. Існує можливість скидання таймера. При своєчасному скиданні таймера відповідний імпульс скидання мікроконтролера блокується. У програмі мікроконтролера у цьому випадку повинні бути реалізовані функції, що забезпечують з заданим часовим проміжком формування імпульсів скидання сторожового таймера.

Іноді супервізори напруги поєднують з сторожовим таймером. Так, наприклад, супервізори MAX823, MAX6316, MAX6318, DS1814/19, DS1832 фірми Dallas Semiconductor обладнані вбудованим сторожовим таймером. На рис.2.7 зображено схему такого супервізора MAX6316.

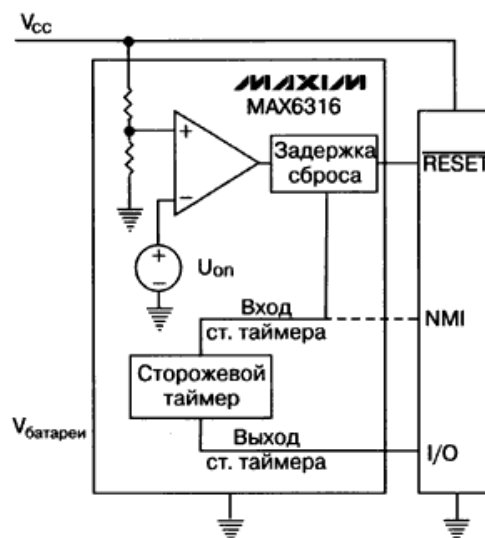


Рис.2.7

2.4. Генератор мікроконтролера

Для формування імпульсів синхронізації мікроконтролера можуть використовуватися наступні засоби:

- Вбудований тактовий генератор;
- Зовнішнє джерело тактових сигналів.

Схеми внутрішніх тактових генераторів мікроконтролерів виконаних по n-MOS та CMOS технологіям, які зображено на рис.2.8,а та рис.2.8,б суттєво відрізняються.

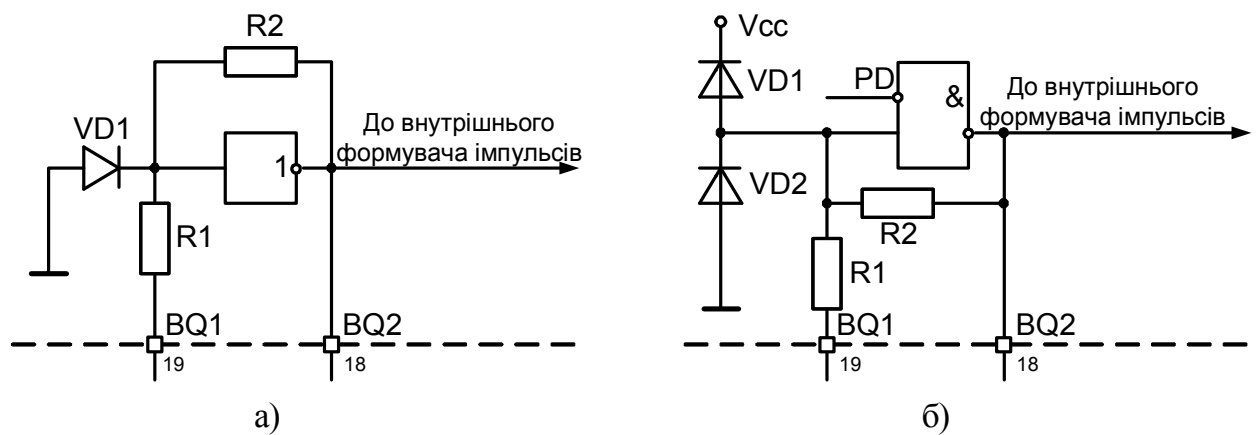


Рис. 2.8

При використанні вбудованого тактового генератора звичайно використовують три типи зовнішніх ланцюгів:

- LC-ланцюг, зображений на рис.2.9,а;
- Керамічний або кварцовий резонатор, що зображені на рис.2.9,б.

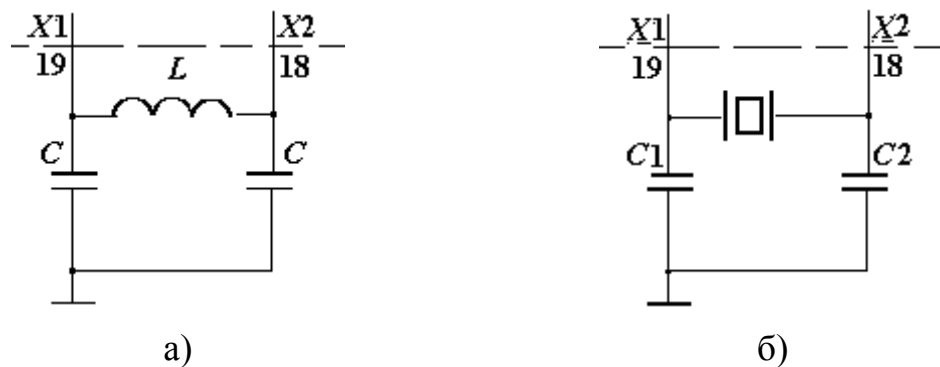


Рис.2.9

Варіант реалізації генератора з зовнішнім LC-ланцюгом має найменшу вартість, може працювати у повному температурному діапазоні мікроконтролера, проте стабільність формованої частоти вкрай низька та суттєво залежить від робочої температури, напруги живлення мікроконтролера, старіння індуктивності та ємностей. Частота імпульсів генератора у цьому випадку визначається наступним чином

$$f = \frac{1}{2\pi\sqrt{LC'}}.$$

Керамічні резонатори мають більшу вартість, проте забезпечують і більшу стабільність частоти. Наприклад, керамічні резонатори фірми Murata CSTCC4.00MG-TC мають точність налаштування частоти $\pm 0,5\%$, стабільність $\pm 0,3\%$ та можуть працювати у звуженому, по відношенню до LC-ланцюгів, температурному діапазоні $-20 \div +80^\circ\text{C}$.

Найкращими показниками по точності та стабільності частоти відзначаються мікропроцесорні кварцові резонатори. Наприклад, резонатори HC49 відзначаються точністю налаштування частоти $\pm 30\text{ppm}$ та стабільністю частоти $\pm 50\text{ppm}$, проте можуть працювати у досить вузькому температурному діапазоні $-10 \div +60^\circ\text{C}$. Їх вартість у декілька разів перевищує вартість керамічних резонаторів.

У документації на керамічні чи кварцові резонатори приводяться значення загальних величин зовнішніх ємностей, що підключаються в схемі генератора (рис.2.9,б). При визначенні номіналів ємностей, що використовуються у схемі (рис.2.9,б), слід враховувати також власну ємність виводів мікроконтролера. Для DIP- корпусів мікроконтролерів така ємність складає близько 12pF, для SMD- корпусів значення ємностей виводів близько 8pF.

Зовнішнє джерело тактових сигналів використовується у тих випадках коли необхідно синхронізувати роботу мікроконтролера з іншими системами схеми, або досягти досить підвищеної точності та стабільності частоти. На рис.2.10,а-б зображено схеми включення зовнішнього джерела синхросигналів при використанні мікроконтролерів виконаних по n-MOS та CMOS технологіям.

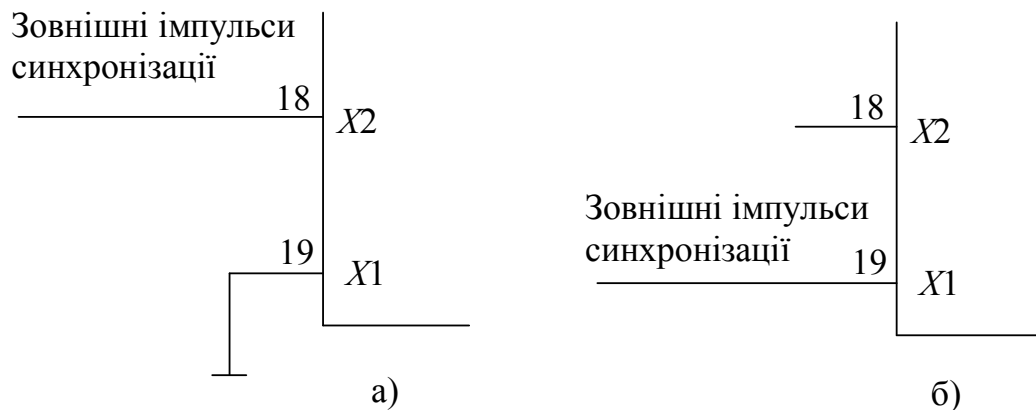


Рис. 2.10

Значна кількість фірм випускає прицевійні контрольовані кварцові генератори. Наприклад, фірма Murata випускає генератори CFPT-141, які мають можливість корекції частоти у межах $\pm 3 \div \pm 7$ ppm, стабільність $\pm 2,5$ ppm та можуть працювати у температурному діапазоні $-30 \div +85^\circ\text{C}$.

2.5. Приклади електричної принципової схеми контролера

У наведених далі прикладах мікроконтролерних пристроїв використовуються не всі елементи. Зокрема, не приведені елементи що забезпечують живлення та скидання мікроконтролера. У всіх прикладах схем використовується генератор мікроконтролера з кварцовим резонатором.

2.5.1. Використання зовнішньої пам'яті програм

Мікроконтролери сімейства MCS51 можуть використовуватися як з внутрішньою, так і зовнішньою пам'яттю програм. Загальна ємність адресованої пам'яті програм мікроконтролера AT89C51 складає 64К.

Для використання внутрішньої пам'яті програм на вивід «EA» мікроконтролера подають логічний рівень «1». У цьому випадку спочатку використовується нижня частина пам'яті програм, що розміщується в мікроконтролері. У разі перевищення об'єму програми розмірів цього сегменту іде звертання до зовнішньої пам'яті.

Для використання лише зовнішньої пам'яті програм на вивід «EA» мікроконтролера подають логічний рівень «0». Структурну схему контролера для цього випадку приведено на рис.2.11.

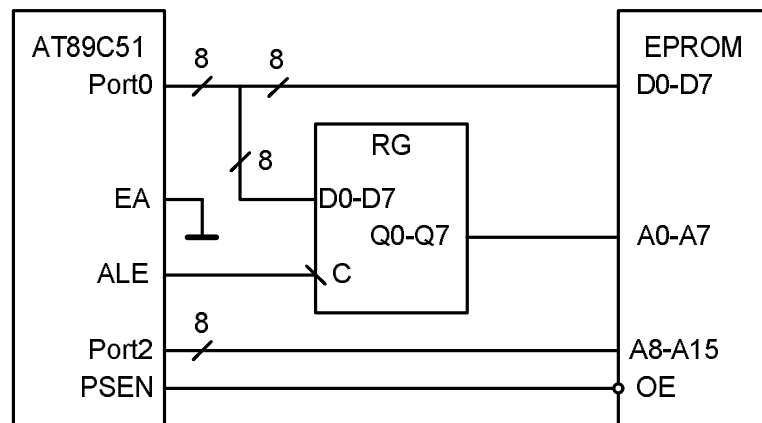


Рис.2.11

Відповідну принципову схему, де ємність зовнішньої пам'яті програм складає 8К, зображено на рис.2.12.

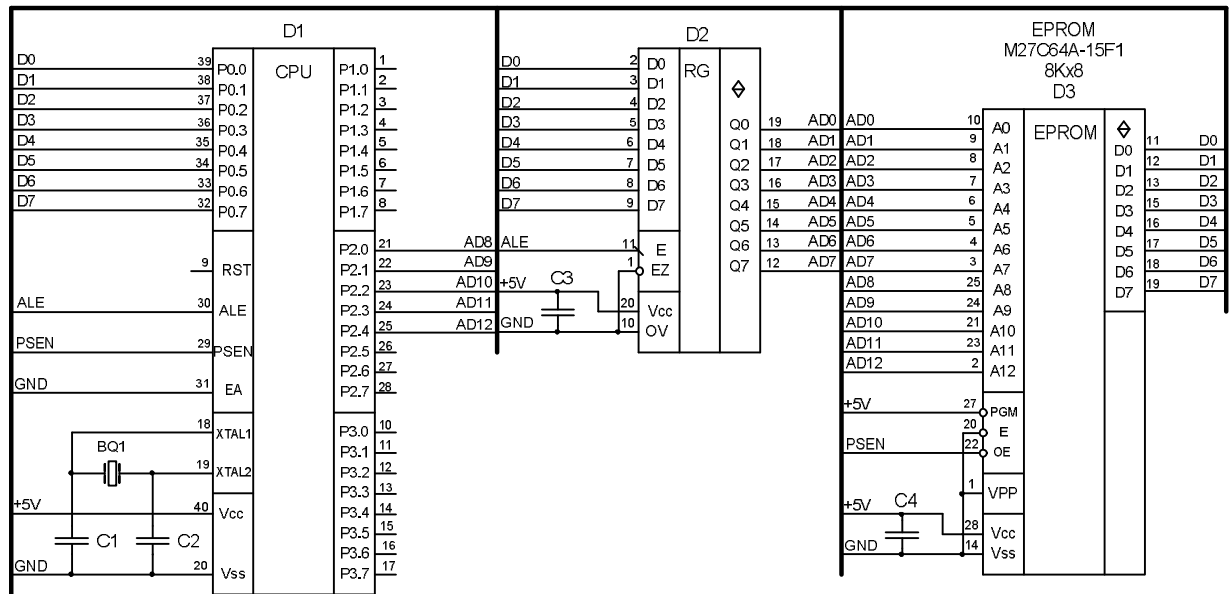


Рис.2.12

2.5.2. Використання зовнішньої пам'яті даних

Мікроконтролери сімейства MCS51 мають можливість роботи зовнішньою пам'яттю даних. Загальна ємність зовнішньої пам'яті даних мікроконтролера AT89C51 складає 64К.

На рис.2.13 приведено структурну схему використання зовнішньої пам'яті даних мікроконтролерів сімейства MCS51.

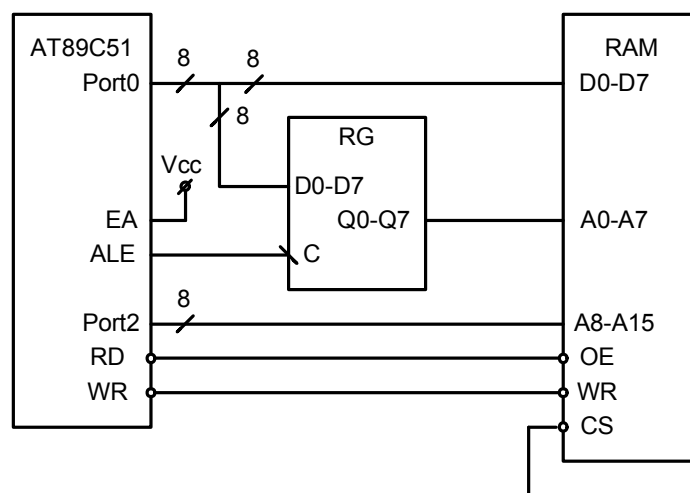


Рис.2.13

Принципову схему, для ємності зовнішньої пам'яті даних складає 32К, зображено на рис.2.14.

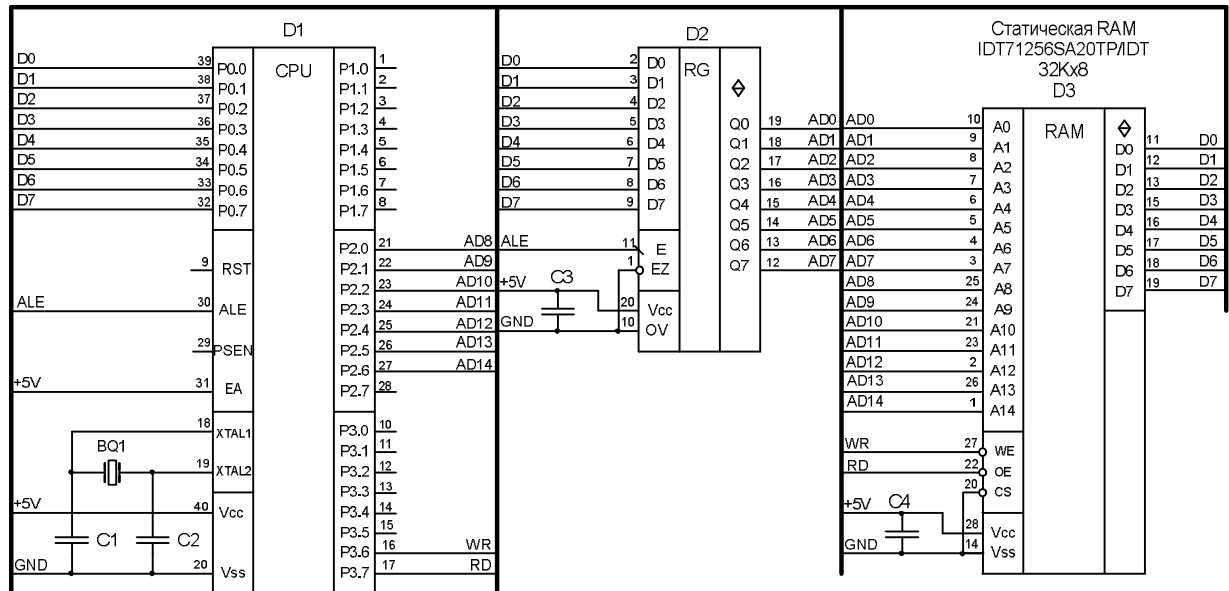


Рис. 2.14

2.5.3. Використання зовнішніх регістрів вводу/виводу даних

Досить часто, з метою розширення кількості портів вводу/виводу цифрових сигналів, у мікроконтролерну систему включають паралельні регістри.

При цьому використовують два варіанти побудови електричних схем:

- З використанням стандартної шини P8, відповідних команд та протоколів обміну з пристроями зовнішньої пам'яті даних;
- З програмною емуляцією шини P8.

На рис.2.15 приведено схему, що реалізує перший підхід. Особливістю цієї схеми є те, адресний простір зовнішньої пам'яті даних використовується не раціонально. У такій реалізації можливе підключення 8 зовнішніх пристроїв. Для адресації цих пристроїв (лінії AD1,AD2...) відводяться сегменти пам'яті даних ємністю 8К. Проте схемотехнічна реалізація контролера виходить досить компактною.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

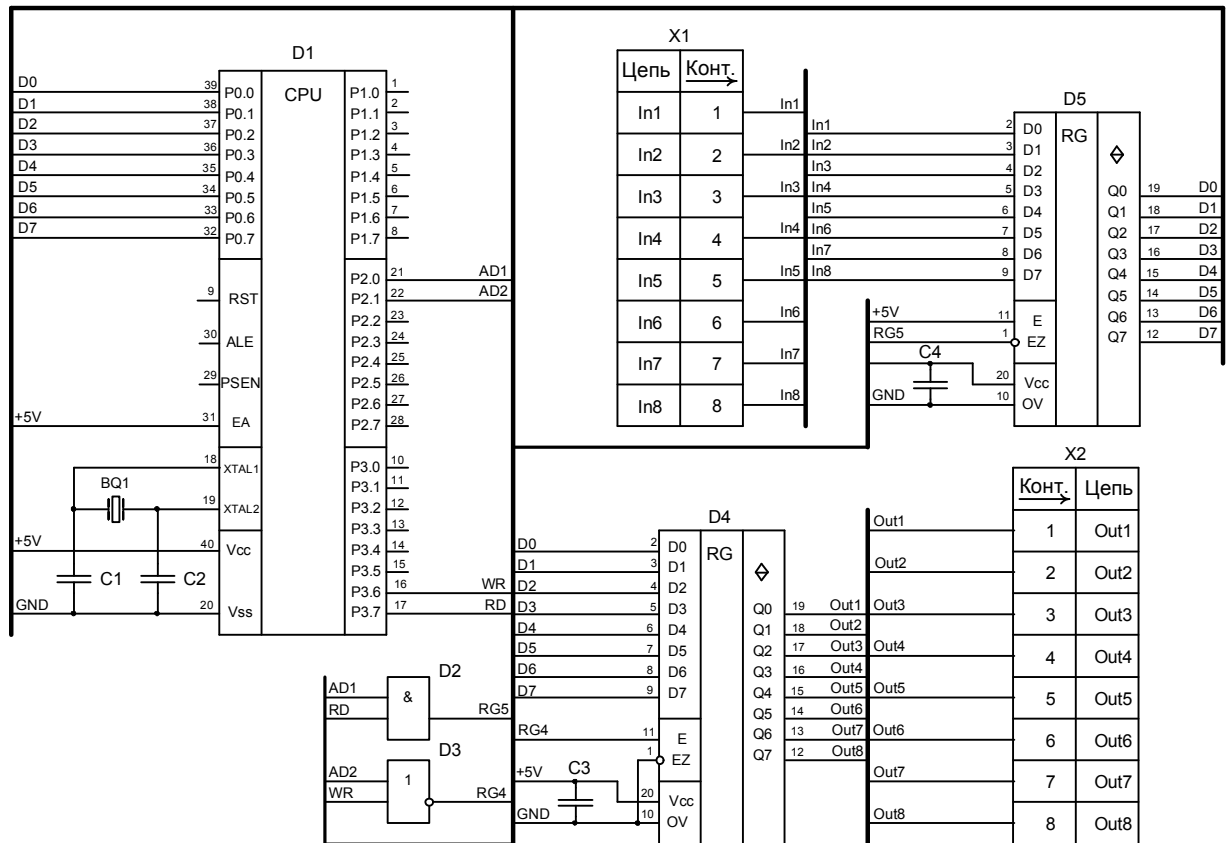


Рис.2.15

3. ОБРОБКА ІНФОРМАЦІЇ

Система команд мікроконтролера надає великі можливості обробки даних, забезпечує реалізацію логічних і арифметичних операцій.

Команди цієї групи дозволяють виконувати наступні операції над 8-бітними цілими двійковими числами: додавання, додавання з урахуванням перенесення, десяткова корекція, інкремент і декремент, віднімання, множення, ділення, диз'юнкція, кон'юнкція, виключаюче АБО, інверсія, скидання і зсув. Опис команд приведено в табл. 2.1, 2.2 та рис. 2.1. У табл. 2.3. наведені умови установки і скидання прапорів.

Ознака паритету Р змінюється будь-якими командами, результат яких змінює акумулятор (включаючи команди пересилки).

Таблиця 2.1. Група команд арифметичних операцій

МНЕМОНІКА	КОП	Б	Ц	ОПЕРАЦІЯ
ADD A,Rn	00101rrr	1	1	$(A):=(A)+(Rn)$
ADD A,ad	00100101	2	1	$(A):=(A)+(ad)$
ADD A,@Ri	0010011I	1	1	$(A):=(A)+((Ri))$
ADD A,#d	00100100	2	1	$(A):=(A)+\#d$
ADDC A,Rn	00111rrr	1	1	$(A):=(A)+(Rn)+(C)$
ADDC A,ad	00110101	2	1	$(A):=(A)+(ad)+(C)$
ADDC A,@Ri	0011011I	1	1	$(A):=(A)+((Ri))+(C)$
ADDC A,#d	00110100	2	1	$(A):=(A)+\#d+(C)$
DA A	11010100	1	1	Десяткова корекція
SUBB A,Rn	10011rrr	1	1	$(A):=(A)-(C)-(Rn)$
SUBB A,ad	10010101	2	1	$(A):=(A)-(C)-(ad)$
SUBB A,@Ri	1001011I	1	1	$(A):=(A)-(C)-((Ri))$

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

SUBB A,#d	10010100	2	1	$(A):=(A)-(C) - \#d$
INC A	00000100	1	1	$(A):=(A)+1$
INC Rn	00001rrr	1	1	$(Rn):=(Rn)+1$
INC ad	00000101	1	1	$(ad):=(ad)+1$
INC @Ri	0000011i	2	1	$((Ri)):=((Ri))+1$
INC DPTR	10100011	1	2	$(DPTR):=(DPTR)+1$
DEC A	00010100	1	1	$(A):=(A)-1$
DEC Rn	00011rrr	1	1	$(Rn):=(Rn)-1$
DEC ad	00010101	1	1	$(ad):=(ad)-1$
DEC @Ri	0001011i	2	1	$((Ri)):=((Ri))-1$
MUL AB	10100100	1	4	$(B)(A):= (A)*(B)$
DIV AB	10000100	1	4	$(B).(A):= (A)/(B)$

Таблиця 3.2. Група команд логічних операцій

МНЕМОНІКА	КОП	Б	Ц	ОПЕРАЦІЯ
ANL A,Rn	01011rrr	1	1	$(A):=(A)\wedge(Rn)$
ANL A,ad	01010101	2	1	$(A):=(A)\wedge(ad)$
ANL A,@Ri	0101011i	1	1	$(A):=(A)\wedge((Ri))$
ANL A,#d	01010100	2	1	$(A):=(A) \wedge \#d$
ANL ad,A	01010010	2	1	$(ad):=(ad)\wedge(A)$
ANL ad,#d	01010011	3	2	$(ad):=(ad)\wedge(\#d)$
ORL A,Rn	01001rrr	1	1	$(A):=(A)\vee(Rn)$
ORL A,ad	01000101	2	1	$(A):=(A)\vee(ad)$
ORL A,@Ri	0100011i	1	1	$(A):=(A)\vee((Ri))$
ORL A,#d	01000100	2	1	$(A):=(A)\vee \#d$
ORL ad,A	01000010	2	1	$(ad):=(ad)\vee(A)$

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

ORL ad,#d	01000011	3	2	(ad):=(ad)∨ #d
XRL A,Rn	01101rrr	1	1	(A):=(A)+(Rn)
XRL A,ad	01100101	2	1	(A):=(A)+(ad)
XRL A,@Ri	0110011i	1	1	(A):=(A)+((Ri))
XRL A,#d	01100100	2	1	(A):=(A)+ #d
XRL ad,A	01100010	2	1	(ad):=(ad)+(A)
XRL ad,#d	01100011	3	2	(ad):=(ad)+ #d
CLR A	11100100	1	1	(A):= 0
CPL A	11110100	1	1	(A):= інверсія (A)
RL A	00100011	1	1	Вліво циклічно (рис. 3.1,а)
RLC A	00110011	1	1	Вліво через C (рис.3.1,б)
RR A	00000011	1	1	Вправо циклічно (рис.3.1,в)
RRC A	00010011	1	1	Вправо через C (рис.3.1,г)
SWAP A	11000100	1	1	Обмін тетрадами (рис.3.1,д)

Організація зсувів

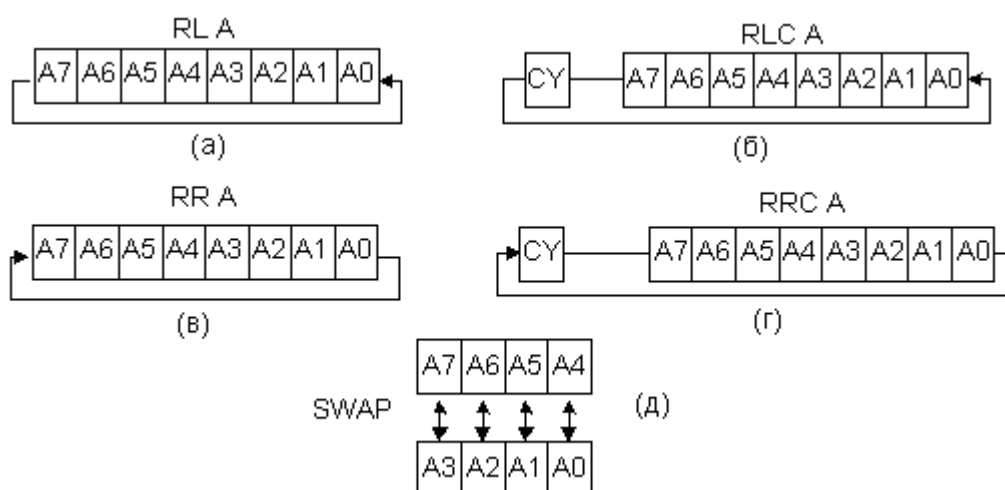


Рис. 3.1.

Таблиця 3.3. Установка прапорів

МНЕМОНІКА	CY	OV	AC
ADD	+	+	+
ADDC	+	+	+
SUBB	+	+	+
MUL	0	+	-
DIV	0	+	-
DA	+	-	-
RRC	+	-	-
RLC	+	-	-
(+) – змінюється; (-) - не змінюється			

МНЕМОНІКА	CY	OV	AC
SET C	1	-	-
CLR C	0	-	-
CPL C	+	-	-
ANL C, bit	+	-	-
ANL C, / bit	+	-	-
ORL C, bit	+	-	-
ORL C, / bit	+	-	-
MOV C, bit	+	-	-
CJNE	+	-	-

Фрагменти програм з використанням арифметичних команд.

1. Операція додавання

MOV A, #34H ; запис в акумулятор числа 34
 MOV R1, #2H ; запис в регістр числа 2
 ADD A, R1 ; додавання 34H та 2H. В акумуляторі результат - 36H.

2. Операція віднімання

MOV A, #34H ; запис в акумулятор числа 34H
 MOV R1, #2H ; запис в регістр R1 числа 2H
 SUBB A, R1 ; віднімання 34H та 2H і прапора C. В акумуляторі - 32H.

3. Операція множення

MOV A, #34H ; запис в акумулятор числа 34
 MOV B, #2H ; запис в регістр числа 2
 MUL AB ; множення 2H*34H, в акумулятор запис результату - 68H.

4. Операція ділення

MOV A,#34 ; запис в акумулятор числа 34
MOV B,#2 ; запис в регістр числа 2
DIV AB ; ділення 34H + 2H. В акумулятор запис результату: 1A.

Фрагменти програм з використанням логічних команд.

1. Логічне „І”

MOV R1, #01010101B ; запис числа в регістр
MOV A, #10101010B ; запис числа в акумулятор
ANL A, R1 ; логічне „і” акумулятора і регістра. Результат 00000000B.

2. Логічне „виключаюче або”

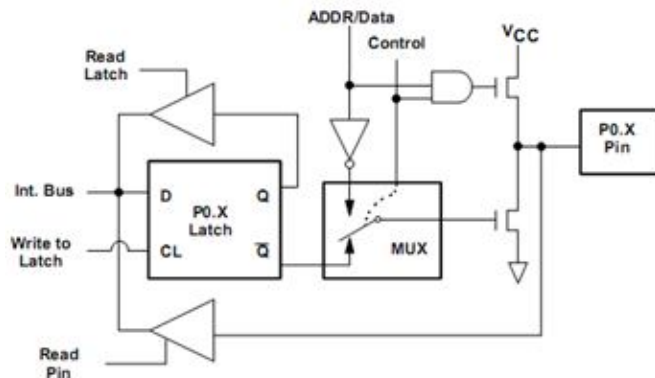
MOV R1, #01010101B ; запис числа в регістр
MOV A, #10101010B ; запис числа в акумулятор
XRL A, R1 ; логічне „ виключаюче або”. Результат 11111111B.

Фрагменти програм з використанням команд зсуву:

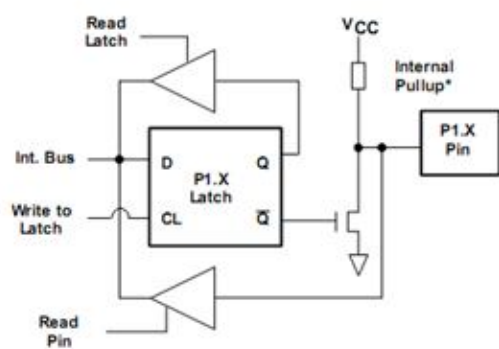
MOV A, #00001111B ; запис числа в акумулятор
SWAP A ; обмін тетрадами в акумуляторі. Результ 11110000B.
RL A ; зсув вліво. Результат 1110001.

4. ПРОГРАМУВАННЯ ПАРАЛЕЛЬНИХ ПОРТІВ

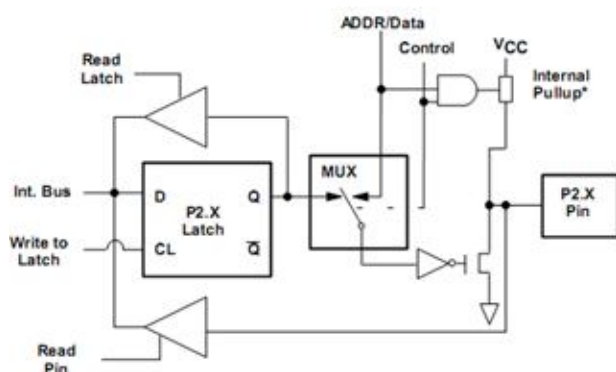
Мікроконтролер AT89C51 має чотири 8 - бітні паралельні двонаправлені порти P0 - P3, які дозволяють вводити і виводити цифрові сигнали. На рис.4.1,а – 4.1, д зображено бітові структурні схеми цих портів.



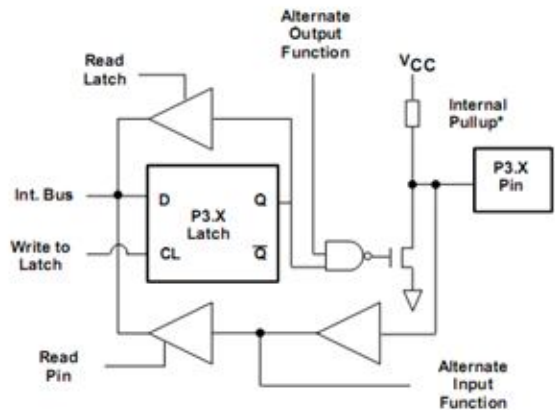
а) Бітова структура порту P0



б) Бітова структура порту P1



в) Бітова структура порту P2



г) Бітова структура порту P3

Рис.4.1

Кожен з портів містить восьмирозрядний регістр, що має байтову і бітову адресацію для установки (запис '1') або скидання (запис '0') розрядів цього регістра за допомогою програмного забезпечення. Виходи цих регістрів з'єднані з драйверами портів. Кожен розряд регістра порту є D-тригером. Дані з внутрішньої шини мікроконтролера записуються в регістр по сигналу "запис". Вихід "Q" D-тригера передається до внутрішньої шини за сигналом "читання". Зна-

чення сигналу безпосередньо з зовнішнього виводу порту зчитується за сигналом "читання виводу".

4.1. Ініціалізація портів

При записі в розряд SFR- регістра порту логічного «0» вихідний транзистор драйвера відкривається і на виводі мікросхеми з'являється низький потенціал, змінити який ззовні неможливо. Якщо в цей розряд записати логічну «1», то вихідний транзистор драйвера закривається і на виводі мікросхеми з'являється високий потенціал. Цей потенціал можливо змінити зовнішніми пристроями на нульовий, наприклад, замиканням цієї ніжки мікросхеми на корпус. В цьому випадку, логічний стан, що зчитується мікроконтролером, буде відповідати стану на виході зовнішнього пристрою. Тому, перед тим як здійснити введення інформації по якомусь виводу порту, відповідний розряд SFR-регістра порту необхідно налаштувати на режим введення - записати в нього логічну «1».

При ініціалізації ліній порту на режим передавача необхідно у відповідні розряди SFR-регістру порту записати значення, що відповідають вихідним значенням передавача. Наприклад, якщо необхідно встановити на лінії порту високий рівень напруги, то до відповідного розряду SFR- регістру цього порту необхідно записати логічну «1».

З тієї ж причини при налаштуванні виводів порту на виконання альтернативних функцій у відповідні розряди повинні бути записані логічні '1'.

При скиданні мікроконтролера в усі розряди SFR- регістрів портів записуються логічна «1», тобто всі порти переходять у режим приймача.

Окрім роботи в якості звичайного порту введення - виведення, порт P3 може виконувати ряд додаткових (альтернативних) функцій.

Кожна лінія порту P3 має індивідуальну альтернативну функцію перелік яких приведено в табл. 4.1.. Лінії порту P3 можуть виконувати альтернативні

функції тільки в тому випадку, якщо у відповідні цим лініям розряди SFR- регістра P3 записані логічні «1».

Таблиця 4.1.

Символ біту	Позиція	Призначення
RD	P3.7	Читання. Активний сигнал низького рівня формується апаратно при зверненні до ВПД
WR	P3.6	Запис. Активний сигнал низького рівня формується апаратно при зверненні до ВПД
T1	P3.5	Вхід таймера/лічильника TC1
T0	P3.4	Вхід таймера/лічильника TC0
INT1	P3.3	Вхід запиту переривання INT1.
INT0	P3.2	Вхід запиту переривання INT0.
TXD	P3.1	Вихід передавача послідовного порту. Вихід синхронізації в режимі регістра зсуву.
RXD	P3.0	Вхід приймача послідовного порту. Введення/вивід даних в режимі регістра зсуву.

Приклад 4.1

Провести ініціалізацію портів на наступні режими роботи:

- P0 – всі лінії порту є приймачами;
- P1 – всі лінії порту є передавачами, початковий стан ліній 00001111b;
- P2 - всі лінії порту є приймачами;
- P3 – старша тетрада порту працює в режимі альтернативних функцій, молодша тетрада знаходиться в режимі передавачів, на виходах яких логічний «0».

```

;-----
ini_P0 equ    11111111b    ;константа ініціалізації порту P0
ini_P1 equ    00001111b    ;константа ініціалізації порту P1
ini_P2 equ    11111111b    ;константа ініціалізації порту P2
ini_P3 equ    11110000b    ;константа ініціалізації порту P3
;-----
        org      0H                ; адреса вектора рестарту після пуску
INIT:                                ; мітка, що відповідає адресі блоку ініціалізації
        mov     P0,#ini_P0        ;передача в порт P0 константи ініціалізації
        mov     P1,#ini_P1        ;передача в порт P1 константи ініціалізації
        mov     P2,#ini_P2        ;передача в порт P2 константи ініціалізації
        mov     P3,#ini_P3        ;передача в порт P3 константи ініціалізації
        end                                ; директива асемблеру про закінчення програми
;-----

```

4.2. Робота з портами

Деякі команди читання порту використовують сигнал "читання засувки", інші "читання виводу".

Читання інформації з виводів мікросхеми здійснюється командами:

- MOV A, P3 ;передати дані з ліній порту P3 в акумулятор;
- JB P3.4,M1;якщо на лінії P3.4 логічна 1,то перейти на мітку M1.

Читання регістра - засувки здійснюється командами читання - модифікація – запис:

- CPL P3.1; проінвертувати сигнал на виводі порту P3.1;
- ORL P2, #10000001b ;Встановити логічну “1” на лініях 7,0 порту P2;
- ANL P3,#11111100b ;Встановити логічний “0” на лініях 0,1 порту P3.

Для запису числа в порт можна скористатися командами для роботи з байтовими даними, що встановлюють логічні рівні одразу на всіх виводах портів:

1. Пересилання (команда MOV).

MOV P2, # 11110000b ;встановити “1” на виводах P2.4 – P2.7 та “0”на решті
;ліній порту P2

MOV P1, # 01110011b ;передати на порт P1 число 01110011b

MOV P3, A ;передати на порт P3 вміст акумулятора;

2. Логічне «І» (команда ANL).

ANL P1, # 11110011b ;встановити “0” на ніжках P1.2 і P1.3

3. «Виключаюче АБО» (команда XRL).

XRL P3, # 01000100b ;інвертувати логічні рівні на виводах P3.2 і P3.6

4. Логічне «АБО» (команда ORL).

ORL P1, # 00001100b ; встановити “1” на ніжках P1.2 і P1.3

Для зміни логічних рівней на окремих виводах портів можна скористатися командами з бітовою адресацією:

1. Пересилання (команда MOV).

5. ПРОГРАМУВАННЯ ТАЙМЕРІВ-ЛІЧИЛЬНИКІВ

В мікроконтролерах сімейства MCS-51 звичайно є два 16-бітних таймера/лічильника T/C0 і T/C1, які можуть бути використані як в якості таймерів, так і в якості лічильників зовнішніх подій. Кожен з них складається з двох 8-бітних регістрів даних TH0 (старший байт) і TL0 (молодший байт) для таймера TC0 або TH1 і TL1 для таймера TC1.

У режимі таймера вміст цих регістрів інкрементується в кожному машинному циклі, через кожні 12 періодів коливань кварцового резонатора.

У режимі лічильника вміст регістрів інкрементується у разі переходу з «1» в «0» зовнішнього вхідного сигналу, що подається на виводи мікроконтролера T0 або T1. Так як на розпізнавання такого переходу потрібні два машинних цикли, максимальна частота підрахунку вхідних сигналів дорівнює 1/24 частоти резонатора. Для гарантованого прочитання вхідний сигнал повинен утримувати значення «0» та «1», як мінімум, протягом одного машинного циклу мікроконтролера.

Таймер/лічильник TC1 також використовується для встановлення швидкості передачі послідовного порту.

Структурні схеми таймерів/лічильників TC0 і TC1, за винятком одного режиму, ідентичні. Таймер/лічильники можуть включатися і виключатися як програмно, за допомогою біт керування TR0/TR1 (за умови GATE0/GATE1 = 0), так і апаратно, шляхом подачі керуючого сигналу на входи INT0/INT1 (за умови одночасного виконання GATE0/GATE1 = 1 та TR0/TR1 = 1). Слід зазначити, що після включення і після скидання мікроконтролера роботу таймерів/лічильників.

Для роботи з цими таймерами/лічильниками використовуються два регістри керування TMOD та TCON.

На рис. 5.1 представлено бітову структуру регістру TMOD. Призначення окремих біт керування описано в табл. 5.1. Всі біти цього регістру доступні як

для читання, так і для запису. При включенні мікроконтролера до регістру TMOD записується нульове значення.

7	6	5	4	3	2	1	0
GATE1	C/T1	M1	M0	GATE0	C/T0	M1	M0

Рис.5.1

Таблиця 5.1.

Біт	Позиція	Призначення
GATE1	TMOD.7	Управління включенням таймера TC1. «1» – зовнішнє включення. Таймер працює якщо одночасно виконуються умови INT1 = 1 і TR1 = 1. «0» – програмний запуск. Таймер працює якщо TR1 = 1.
C/ \overline{T} 1	TMOD.6	Біт вибору типу роботи для таймера TC1: «0» - працює як таймер; «1» - працює як лічильник.
M1.1	TMOD.5	Визначення режиму роботи TC1.
M1.0	TMOD.4	Визначення режиму роботи TC1.
GATE0	TMOD.3	Управління включенням таймера TC0. «1» – зовнішнє включення. Таймер працює якщо одночасно виконуються умови INT0 = 1 і TR0 = 1. «0» – програмний запуск. Таймер працює якщо TR0 = 1.
C/ \overline{T} 0	TMOD.2	Біт вибору типу роботи для таймера TC0: «0» - працює як таймер; «1» - працює як лічильник.
M0.1	TMOD.1	Визначення режиму роботи TC0.
M0.0	TMOD.0	Визначення режиму роботи TC0.

На рис. 5.2 представлено бітову структуру регістру TCON. У табл. 5.2 описано призначення окремих біт керування. Для контролю стану та керування режимами роботи таймерів/лічильників використовуються старші чотири біти цього регістру – TF1, TR1, TF0, TR0. Біти керування молодшої тетради відповідають за налаштування режимів роботи зовнішніх переривань по входах INT0 та INT1.

Всі біти цього регістру доступні як для читання, так і для запису. При включенні мікроконтролера до регістру TCON записується нульове значення.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Рис.5.2

Таблиця 5.2.

Біт	Позиція	Призначення
TF1	TCON.7	Прапор переповнення таймера TC1. Встановлюється при переповненні. Скидається при обслуговуванні переривання апаратно.
TR1	TCON.6	Біт запуску/зупинки таймера TC1. Встановлюється/скидається програмно.
TF0	TCON.5	Прапор переповнення таймера TC0. Встановлюється при переповненні. Скидається при обслуговуванні переривання апаратно.
TR0	TCON.4	Біт запуску/зупинки таймера TC0. Встановлюється/скидається програмно.
IE1	TCON.3	Прапор зовнішнього переривання по входу INT1. Встановлюється апаратно відповідним сигналом на вході INT1. Скидається апаратно при обслуговуванні переривання, або програмно.
IT1	TCON.2	Біт управління типом сигналу, що викликає переривання: IT1 = 0 – переривання по низькому рівню сигналу; IT1 = 1 – переривання по зрізу сигналу. Встановлюється/скидається програмно.
IE0	TCON.1	Прапор зовнішнього переривання по входу INT0. Встановлюється апаратно відповідним сигналом на вході INT0. Скидається апаратно при обслуговуванні переривання, або програмно.
IT0	TCON.0	Біт управління типом сигналу, що викликає переривання: IT1 = 0 – переривання по низькому рівню сигналу; IT1 = 1 – переривання по зрізу сигналу. Встановлюється/скидається програмно.

Обидва таймера/лічильника можуть налаштовуватися для роботи в чотирьох режимах:

- режим 0 - 13-бітний таймер;
- режим 1 - 16-бітний таймер;
- режим 2 - 8-бітний автоперезавантажувальний таймер;
- режим 3 - таймер TC0, як два роздільних 8-бітних таймера.

У табл. 5.3 приведено відповідність між станом біт керування M0.0, M0.1 (M1.0, M1.1) та режимами роботи таймерів/лічильників.

Таблиця 5.3

Режим	M1	M0	Опис режиму роботи
0	0	0	13-бітний таймер/лічильник
1	0	1	16-бітний таймер/лічильник
2	1	0	8-бітний таймер/лічильник з автоперезавантажуванням
3	1	1	ТС0, як два 8-бітних таймера/лічильник

Режим «0».

Цей режим був введений для сумісності із сімейством мікроконтролерів MCS48 для полегшення перенесення вже розроблених програм на мікроконтролери сімейства MCS51. У нових розробках цей режим практично не використовується.

У нульовому режимі таймери\лічильники тотожні і працюють як 13-бітні сумуючі лічильники. Кожний з таймерів\лічильників складається з 8 бітного регістра TH0/ TH1 і молодших 5 біт регістра TL0/TL1. Старші 3 біта регістрів TL0/TL1 не визначені і ігноруються.

Регістри TL0/TL1 працюють як попередні подільники на 32. Коефіцієнт ділення попереднього подільника можна модифікувати шляхом циклічного завантаження необхідної константи. Проте у цьому випадку виникають суттєві складності по поточному контролю їх стану та перезавантаженню. Доцільно використовувати попередній подільник з коефіцієнтом ділення на 32. Для налаштування необхідного коефіцієнту застосовувати регістри TH.

На рис.5.3 зображено структурну схему першого таймера/лічильника в «0» режимі роботи.

У разі переповнення регістрів TH1, TL1 лічильника встановлюється (приймає значення "1") прапор переривання таймера TF1.

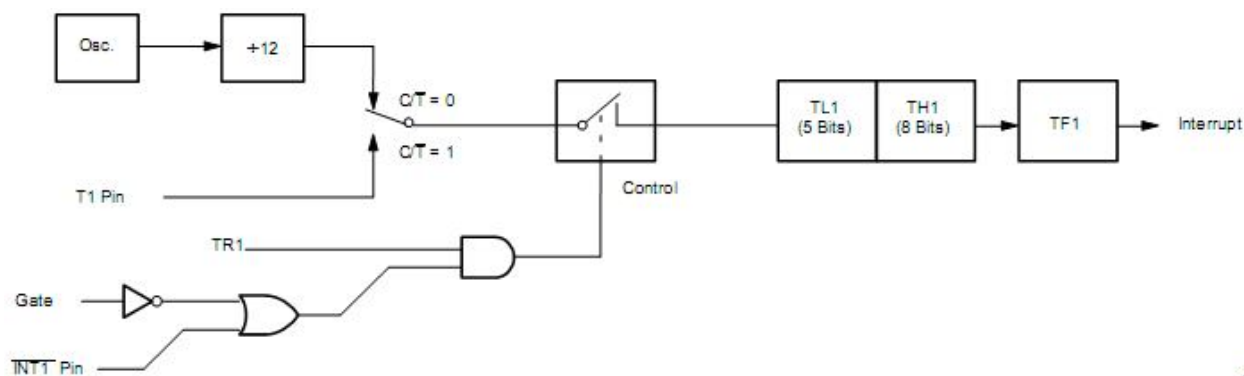


Рис.5.3

Для налаштування режимів роботи «таймер» або «лічильник» використовується біт керування С/Т. У разі запису у цей біт «0» на регістри TH1/TL1 поступають імпульси циклу і таймер/лічильник працює як таймер. При запису у цей біт «1» підраховуються імпульси, що поступають на вивід мікроконтролера T1. У цьому випадку таймер/лічильник працює як лічильник.

Згідно з структурною схемою можливі два варіанта включення таймера/лічильника – програмне та зовнішнє керування. Програмне керування включенням забезпечується за рахунок скидання біту GATE1=0 та встановлення біту TR1=1. Зовнішнє керування реалізується при встановленні біту GATE1=1. Таймер/лічильник буде включено, якщо одночасно будуть виконані умови TR1=1 та INT1=1.

Слід зауважити, що у разі роботи таймера/лічильника з зовнішнім керуванням включенням, або в режимі лічильника, необхідно переводити від повідні лінії портів (INT1, T1) у режим альтернативних функцій.

Приклад 5.1.

Налаштувати таймер/лічильник TC0 на «0» режим роботи в якості таймера з зовнішнім управлінням. Встановити значення коефіцієнта ділення 512. Увімкнути таймер/лічильник.

При визначенні констант завантаження регістрів TH0, TL0 виходимо з того, що регістр TL0 працює в режимі попереднього подільника з коефіцієнтом ділення 32. Тоді регістр TH0 має забезпечити коефіцієнт ділення $512/32=16$.

Константу завантаження регістра TMOD для цього завдання визначають у відповідності з даними, що приведено в табл.5.4.

Таблиця 5.4

	Таймер/лічильник 1				Таймер/лічильник 0			
№ біта	7	6	5	4	3	2	1	0
Біт	0	0	0	0	1	0	0	0
Функція	Програмне управління включенням	Режим таймера	«0» режим роботи		Управління включенням по входу INT0	Режим таймера	«0» режим роботи	

Текст програми має наступний вигляд.

```

;-----
;--- Опис констант та змінних
INI_TMOD EQU 00001000b ;константа завантаження регістру TMOD
INI_TH0 EQU 16 ;константа завантаження регістру TH0
;--- Програма
ORG 0H ;адреса рестарту після пуску процесора
SJMP START ;перехід на початок основної програми
ORG 20H ;початкова адреса основної програми
START:
MOV TMOD, #INI_TMOD ;ініціалізація регістру TMOD
MOV TL0, #INI_TH0 ;завантаження регістра TL0
SETB INT0 ; дозвіл режиму альтернативних функцій
;по входу зовнішнього управління TC0
SETB TR0 ;ввімкнення таймера/лічильника TC0
END ;директива про закінчення програми
;-----

```

Режим «1»

В першому режимі роботи таймер/лічильник працює як шістнадцятиригідний лічильник. Режим «1» схожий на режим «0», за винятком того, що в регістрах TH, TL використовуються всі 16 біт.

В цьому режимі можна забезпечити формування інтервалу часу тривалістю до 65536 мкс при частоті генератора мікроконтролера 12 МГц.

Нульовий і перший режими роботи таймерів T0 і T1 призначені для формування одиничного інтервалу часу. Якщо виникає необхідність формувати послідовність інтервалів часу для періодичних процесів, то перезавантаження регістрів TH0 і TL0 для завдання потрібного інтервалу часу проводиться програмно, що для коротких інтервалів часу може призвести до значних витрат процесорного часу.

Приклад 5.2.

Налаштувати перший таймер/лічильник на «1» режим роботи лічильником з внутрішнім управлінням. Константа завантаження – 1A1Fh. Увімкнути таймер/лічильник.

Константу завантаження регістра TMOD для цього завдання визначають у відповідності з даними, що приведено в табл.5.5.

Таблиця 5.5

	Таймер/лічильник 1				Таймер/лічильник 0			
№ біта	7	6	5	4	3	2	1	0
Біт	0	1	0	1	0	0	0	0
Функція	Програмне управління включенням	Режим лічильника	«1» режим роботи		Програмне управління включенням	Режим таймера	«0» режим роботи	

Текст програми має наступний вигляд.

```

;-----
;--- Опис констант та змінних
INI_TMOD EQU 01010000b ;константа завантаження регістру TMOD
INI_THTL EQU 1A1Fh ;константа завантаження регістрів TH1,TL1
;--- Програма
ORG 0H ;адреса рестарту після пуску процесора
SJMP START ;перехід на початок основної програми
ORG 20H ;початкова адреса основної програми
START:
MOV TMOD,#INI_TMOD ;ініціалізація регістру TMOD
MOV TH1, #high(INI_THTL) ;завантаження регістра TH1
MOV TL1, #low (INI_THTL) ;завантаження регістра TL1
SETB T1 ;дозвіл альтернативних функцій по входу T1
SETB TR1 ;увімкнення таймера/лічильника TC1
END ;директива про закінчення програми
;-----

```

Режим «2»

Другий режим використовують для формування послідовності однакових інтервалів часу. У цьому режимі роботи таймери/лічильники TC0 та TC1 є 8-бітовими та працюють однаково.

На рис.5.4 зображено структурну схему першого таймера/лічильника у «2» режимі роботи.

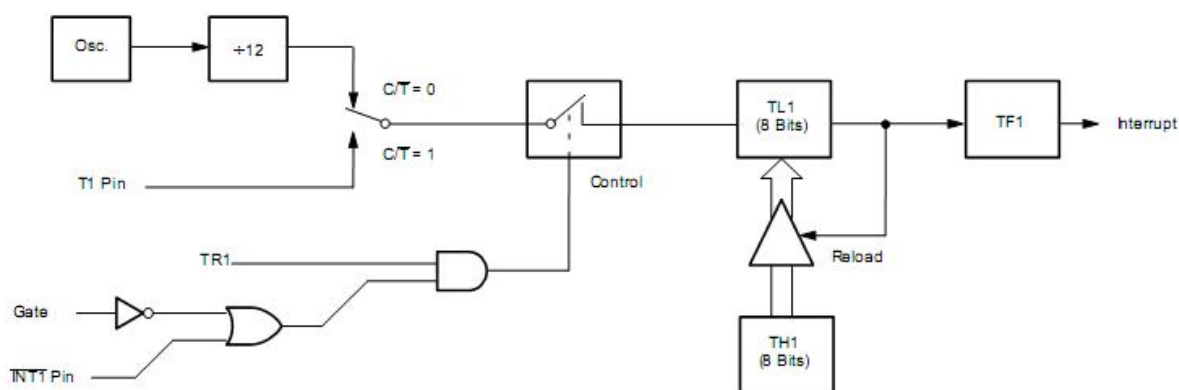


Рис.5.4

Структура керуючої частини таймера/лічильника співпадає з відповідними структурами у режимах «1, 2». Суттєво відрізняються частини, що відносяться до використання регістрів TH1 та TL1.

У режимі «2» регістр TL1 працює як 8-бітний лічильник з автоматичним перезавантаженням початкового значення з регістра TH1 у регістр TL1. Переповнення регістра TL1 не тільки встановлює прапор TF1, але і завантажує регістр TL1 вмістом регістра TH1, який попередньо ініціалізується програмно. Перезавантаження не змінює вміст регістру TH1.

Приклад 5.2

Налаштувати таймер/лічильник TC1 на «2» режим роботи таймером з внутрішнім управлінням. Константа завантаження – 0FАН. Увімкнути таймер/лічильник.

Константу завантаження регістра TMOD для цього завдання визначають у відповідності з даними, що приведено в табл.5.6.

Таблиця 5.6

	Таймер/лічильник 1				Таймер/лічильник 0			
№ біта	7	6	5	4	3	2	1	0
Біт	0	0	1	0	0	0	0	0
Функція	Програмне управління включенням	Режим таймера	«2» режим роботи		Програмне управління включенням	Режим таймера	«0» режим роботи	

Текст програми має наступний вигляд.

```

;-----
;--- Опис констант та змінних
INI_TMOD EQU 00100000b ;константа завантаження регістру TMOD
INI_THTL EQU 0FAh      ;константа завантаження TH1, TL1
;--- Програма
ORG 0H                  ;адреса рестарту після пуску процесора
SJMP START              ;перехід на початок основної програми
ORG 20H                 ;початкова адреса основної програми
START:
MOV TMOD, #INI_TMOD     ;ініціалізація регістру TMOD
MOV TH1, #INI_THTL      ;завантаження регістра TH1
MOV TL1, #INI_THTL      ;завантаження регістра TL1
SETB TR1                ;ввімкнення таймера/лічильника TC1
END                      ;директива про закінчення програми
;-----

```

Режим «3»

Робота таймерів/лічильників TC0 та TC1 у «3» режимі суттєво відрізняється. На рис.5.5 зображено структурну схему таймерів/лічильників у цьому режимі роботи.

Таймер/лічильник TC1 в режимі «3» не працює, він просто зберігає значення регістрів лічильника TH1 та TL1. Проте він може працювати в режимах «0 - 2», якщо таймер/лічильник TC0 знаходиться в режимі «3».

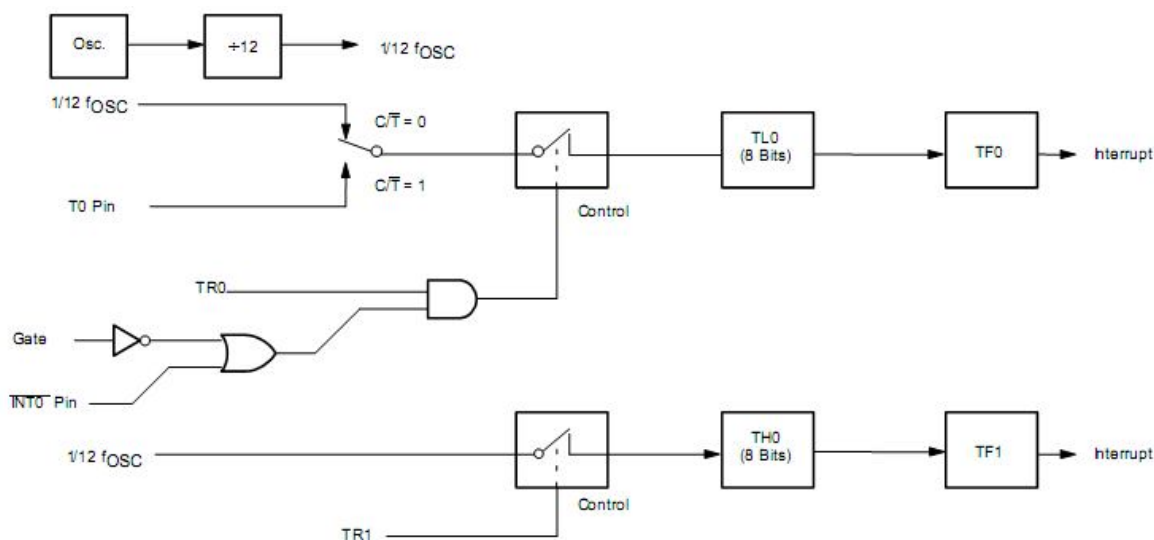


Рис.5.5

Таймер/лічильник TC0 в режимі «3» представляє собою два роздільних 8 бітних лічильника побудованих на основі регістрів TL0 і TH0. Регістр TL0 використовує наступні біти управління таймера TC0: C/T0, GATE0, TR0 і TF0. Таймер/лічильник побудований на основі регістра TH0 працює тільки в режимі таймера і використовує біти керування та контролю TR1 і TF1 таймера TC1. Цей таймер формує переривання по прапорі TF1 таймера TC1.

Логіка роботи таймера TC0 в режимі «3» показана на схемі:

- можливі два варіанта включення таймера/лічильника побудованого на основі регістру TL0 – програмне та зовнішнє керування.

Програмне керування включенням забезпечується за рахунок скидання біту GATE0=0 та встановлення біту TR0=1.

Зовнішнє керування реалізується при встановленні біту GATE0=1. У цьому випадку таймер/лічильник буде включено, якщо одночасно будуть виконані умови TR0=1 та INT0=1;

- дозволяється робота таймера/лічильника побудованого на основі регістру TH0 - якщо біт TR1 = 1;
- при роботі таймера TC0 в режимі «3», таймер TC1 постійно включений у режимах «0 – 2» і виключений при налаштуванні його на режим «3».

6. ПРОГРАМУВАННЯ ПОСЛІДОВНОГО ПОРТУ

Послідовний порт часто називають універсальним асинхронним прийомо-передавачем (УАПП - UART).

Через УАПП здійснюється прийом і передача інформації, представленої послідовним кодом. Порт послідовного введення-виведення використовують в асинхронному режимі для зв'язку зі стандартними периферійними пристроями (дисплеєм, модемом, іншими контролерами). У синхронному режимі УАПП може бути застосований для розширення кількості ліній введення-виведення з використанням зовнішніх зсувних регістрів.

До складу УАПП входять (рис. 6.1):

- зсувний регістр прийому;
- зсувний регістр передачі;
- буферний регістр прийому-передачі SBUF;
- схеми управління та обробки зовнішніх сигналів управління.



Рис. 6.1 Структура послідовного порту

Запис байту у буфер SBUF програмним шляхом призводить до автоматичного переписування байта в зсувний регістр передачі та ініціює початок передачі байта. Наявність буферного регістра прийому дозволяє поєднувати опера-

цію пересилання раніше прийнятого байта у внутрішню пам'ять з прийомом наступного байта. Якщо до моменту закінчення прийому байта попередня інформація не була зчитана з SBUF, вона втрачається.

Управління роботою УАПП здійснюється за допомогою регістра SCON який має побітову адресацію.

При закінченні передачі байту формується сигнал переривання TI основної програми мікроконтролера. При заповненні регістра приймача формується сигнал переривання RI. Ці прапори апаратно не скидаються відповідною програмою переривань, необхідно використовувати програмні засоби.

Бітова структура регістра SCON наведена на рис.6.2, а призначення окремих бітів описано в табл. 6.1.

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Рис. 6.2.

Таблица 6.1

Біт	Позиція	Призначення
SM0	SCON.7	Біт 0 визначення режиму роботи послідовного порту.
SM1	SCON.6	Біт 1 визначення режиму роботи послідовного порту.
SM2	SCON.5	Дозвіл мультипроцесорного обміну. У режимі 0 повинен бути скинутим. У режимі 1 при SM2 = 1 біт RI не встановлюється, якщо прийнятий стоп-біт = 0. У режимах 2 і 3 при SM2 = 1 біт RI не встановлюється, якщо прийнятий дев'ятий біт даних RB8 = 0.
REN	SCON.4	Дозвіл прийому. При REN = 1 прийом дозволено.
TB8	SCON.3	9-біт даних, який буде передаватись у режимах 2 і 3. Значення встановлюється програмно за розсудом розробника.
RB8	SCON.2	У режимі 0 не використовується. У режимі 1 при SM2 = 0 у нього заноситься стоп-біт. У режимах 2 і 3 в нього заноситься 9-й прийнятий біт даних.
TI	SCON.1	Прапор запиту переривань передавача. Повинен скидатися програмно
RI	SCON.0	Прапор запиту переривань приймача. Повинен скидатися програмно

У табл. 6.2 приведено відповідність між режимами УАПП, станом біт керування SM0 і SM1 та основними характеристиками режимів роботи.

Таблиця 6.2

Режим	SM1	SM0	Швидкість передачі	Опис режиму роботи
0	0	0	$f_{osc}/12$	Зсувний регістр
1	0	1	змінна	Асинхронний прийомо-передавач, 8 біт даних у кадрі
2	1	0	$f_{osc}/64$, або $f_{osc}/32$	Асинхронний прийомо-передавач, 9 біт даних у кадрі
3	1	1	змінна	Асинхронний прийомо-передавач, 9 біт даних у кадрі

Швидкість прийому-передачі залежить від режиму роботи УАПП і в режимах 1 ... 3 визначається з урахуванням значення розряду PCON.7 регістра керування потужністю PCON (байт доступний регістр). Цей розряд називається SMOD. При значенні SMOD = 1 - швидкість передачі подвоюється.

Режими роботи УАПП

Синхронний обмін (режим «0»)

При синхронному обміні кожен біт даних супроводжується передачею сигналу синхронізації.

Режим «0»

Дані передаються та приймаються по лінії RxD. По лінії TxD передаються сигнали синхронізації, які формує мікроконтролер.

Інформація видається молодшими бітами вперед.

Швидкість передачі біт залежить від тактової частоти резонатора (f_{osc}) та визначається за наступною формулою

$$BR = f_{osc}/12.$$

Передача даних ініціюється командою передачі байту даних до регістру SBUF, наприклад, MOV SBUF, A. Після цього байт інформації автоматично пе-

реписується в зсувний регістр передачі, а схема управління генерує синхроімпульси зсуву, які надходять на зсувний регістр передачі і на лінію порту TxD. При цьому вміст регістру передачі зсувається вправо і черговий біт інформації надходить на лінію порту RxD. При завершенні передачі схема управління встановлює прапор TI (SCON.1).

Прийом даних починається при встановленні дозволу роботи приймача (REN = 1), якщо був знятий запит на переривання, сформований на попередньому циклі прийому (тобто RI = 0). Після програмної установки біта REN блок управління включає схему формування синхроімпульсів зсуву, які надходять синхронно на зсувний регістр прийому та на лінію порту TxD.

В зсувному регістрі прийому починає формуватися байт даних, значення біт якого зчитуються з входу RxD. Після завершення прийому останнього біту даних вміст зсувного регістру переписується в SBUF і встановлюється прапор запиту переривання RI (SCON.0).

В даному режимі до виводів RxD і TxD мікроконтролера можуть бути підключені зовнішні зсувні регістри. Ці регістри будуть перетворювати послідовний код в паралельний при передачі інформації, або паралельний код в послідовний при прийомі даних мікроконтролером.

Асинхронний обмін (режими «1, 2, 3»).

При асинхронному обміні дані передаються по лінії TxD, а приймаються по лінії RxD. У цьому режимі обміну для узгодження передавач та приймач налаштовують на один і той же протокол обміну.

Режим «1»

У цьому режимі дані передаються через лінію порту TxD та приймаються через лінію RxD. Використовується наступний протокол обміну: передаються та приймаються десять бітів інформації з наступною упаковкою - старт-біт ("0"), 8

біт даних і стоп-біт ("1"). У байті даних використовується спосіб упаковки молодшими бітами вперед.

Швидкість прийому-передачі (BR – baud rate) змінюється програмним шляхом і залежить від значення біта управління PCON.7 (SMOD), і частоти переповнення таймера TC1, яка позначена F_{OVFTC1} :

$$BR = \frac{2^{SMOD}}{32} \cdot F_{OVFTC1}.$$

Переривання від таймера TC1 має бути заблокованим. Таймер/лічильник T/C1 може працювати в будь-якому режимі. Однак найбільш зручним є використання режиму «2» з автоматичним перезавантаження. У цьому випадку швидкість обміну визначається за наступною формулою:

$$BR = \frac{2^{SMOD}}{32} \cdot \frac{f_{osc}}{12 \cdot [256 - (TH1)]}.$$

Передача ініціюється командою передачі даних до регістру SBUF, наприклад, MOV SBUF, A.

Прийом починається після програмної установки біта REN в "1" при виявленні переходу сигналу на RxD з "1" в "0" (старт-біт). У кожному періоді подання біта вхід RxD опитується тричі, і значення біта, який приймається, визначається мажоритарним методом. При прийомі даних значення стоп-біта ("1") фіксується в біт RB8 регістру SCON, і може використовуватись для контролю завершення прийому.

Режим «2»

У цьому режимі дані передаються через TxD та приймаються через RxD. Використовується наступний протокол обміну: передаються та приймаються 11 бітів інформації з наступною упаковкою - стартовий біт ("0"), 8 біт даних, біт контролю, що задається програмно (так званий "дев'ятий" біт) і стоп-біт ("1").

При передачі "дев'ятий" біт TB8 може використовуватися, наприклад, для контролю парності або непарності кількості "1" у байті даних, тобто для переві-

рки достовірності отриманої інформації. При прийомі "дев'ятий" біт надходить в розряд RB8 регістру SCON.

Частота передачі залежить від значення біта SMOD і тактової частоти:

$$BR = \frac{2^{SMOD}}{64} \cdot f_{osc}.$$

Режим «3»

Відрізняється від режиму «2» тільки можливістю програмної зміни швидкості обміну, яка налаштовується так само, як в режимі «1».

Приклад 6.1

Запрограмувати послідовний порт на «0» режим роботи. Встановити швидкість обміну 200000 біт/с.

У цьому режимі роботи SM0= 0, SM1= 0. Константу завантаження регістра SCON приведено в табл.6.3.

Таблиця 6.3

№ біта	7	6	5	4	3	2	1	0
Ім'я біта	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Біт	0	0	0	0	0	0	0	0

Для налаштування заданої швидкості обміну необхідно визначити відповідну частоту резонатора генератора, $F_{osc}=12 \cdot 200000 = 2400000$ Гц.

Текст програми має наступний вигляд.

```

;-----
;--- Опис констант та змінних
INI_P3      equ 00000011b    ;константа ініціалізації порту P3
                                ;дозвіл альтернативних функцій TxD, RxD
INI_SCON     equ 00000000b    ;константа ініціалізації SCON "0-режим"
;--- Програма
            ORG 0H              ;адреса рестарту після пуску процесора
            SJMP INIT           ;перехід на початок основної програми
            ORG 20H             ;початкова адреса блоку ініціалізації
INIT:
            MOV P3,    #INI_P3   ;ініціалізація порту P3
            MOV SCON,  #INI_SCON ;задання режиму роботи UART контролера
            SETB REN             ;включення приймача UART-контролера
            END                  ;директива про закінчення програми
;-----

```

Приклад 6.2

Запрограмувати послідовний порт на «1» режим роботи. Встановити швидкість обміну 2400 біт/с. Частота резонатора генератора складає 12 МГц.

Без використання системи переривань скласти підпрограми реакцій на завершення передачі та прийому байта.

У цьому режимі роботи SM0= 1, SM1= 0. Константу завантаження регістра SCON приведено в табл.6.4.

Таблиця 6.4

№ біта	7	6	5	4	3	2	1	0
Ім'я біта	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Біт	1	0	0	0	0	0	0	0

Для налаштування заданої швидкості обміну необхідно використати таймер/лічильник TC1 та визначити режим роботи та відповідні константи завантаження лічильних регістрів. Згідно з формулою для розрахунку швидкості обміну для «1» режиму роботи UART визначають константу завантаження регістрів таймера/лічильника TC1. При проведенні розрахунку використовують «2» режим роботи таймера/лічильника з автоматичним перезавантаженням та приймають значення біту подвоєння швидкості SMOD=0. Розрахункове значення константи завантаження регістрів складає 243.

У табл.6.5 приведено константу завантаження регістра TMOD, яка відповідає «2» режиму роботи таймера/лічильника TC1.

Таблиця 6.5

№ біта	7	6	5	4	3	2	1	0
Ім'я біта	GATE1	C/T1	M1	M0	GATE0	C/T0	M1	M0
Біт	0	0	1	0	0	0	0	0

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Текст програми має наступний вигляд.

```
;-----  
;--- Опис констант та змінних  
INI_P3      equ    00000011b ;константа ініціалізації порту P3  
                                ;дозвіл альтернативних функцій TxD, RxD  
INI_SCON     equ    10000000b ;SCON – UART у «1» режимі  
INI_PCON     equ    00000000b ;константа ініціалізації PCON, SMOD=0  
INI_TMOD     equ    00100000b ;TMOD – таймер TC1 у «2» режимі  
INI_TC1      equ    243       ;ініціалізація TH1, TL1 – BR=2400 біт/с  
;--- Програма  
    ORG 0H                      ;адреса рестарту після пуску процесора  
    SJMP INIT                   ;перехід на початок основної програми  
    ORG 20H                     ;початкова адреса блоку ініціалізації  
INIT:  
    MOV P3,    #INI_P3          ;ініціалізація порту P3  
    MOV SCON,  #INI_SCON        ;задання режиму роботи UART контролера  
    MOV PCON,  #INI_PCON        ;задання режиму роботи UART контролера  
    MOV TMOD,  #INI_TMOD        ;задання режиму роботи TC1  
    MOV TH1,   #INI_TC1         ;завантаження регістрів таймера TC1  
    MOV TL1,   TH1              ;завантаження регістрів таймера TC1  
    SETB TR1    ;включення таймера/лічильника TC1  
    SETB REN    ;включення приймача UART-контролера  
;--- Головна програма  
MAIN:  
MAIN_RxD:  
    JNB RI, MAIN_TxD            ;перевірка прапора завершення прийому  
    CALL UART_RxD              ;виклик підпрограми обробки прийому  
MAIN_TxD:  
    JNB TI, MAIN                ;перевірка прапора завершення передачі  
    CALL UART_TXD              ;виклик підпрограми обробки передачі  
    SJMP MAIN                   ;заиклення головної програми  
;---Підпрограма обробки завершення прийому  
UART_RxD:  
    CLR RI                      ;скидання прапора переривань приймача  
    RET                         ;повернення з підпрограми  
;---Підпрограма обробки завершення передачі  
UART_TXD:  
    CLR TI                      ;скидання прапора переривань передавача  
    RET                         ;повернення з підпрограми  
  
    END                         ;директива про закінчення програми  
;-----
```

7. ПРОГРАМУВАННЯ СИСТЕМИ ПЕРЕРИВАНЬ

Система переривань - одна з важливих частин мікроконтролера. Переривання припиняє нормальний хід програми для виконання пріоритетного завдання, яке визначається внутрішню або зовнішню подією.

Для кожної такої події розробляється окрема програма, яку називають підпрограмою обробки запиту на переривання (підпрограмою переривання).

При виникненні події, що викликає переривання, мікроконтролер припиняє виконання поточної програми, зберігає у стеку вміст лічильника команд, і переходить до виконання підпрограми обробки переривання. Після виконання підпрограми переривання здійснюється оновлення попередньо збереженого лічильника команд і мікроконтролер повертається до виконання перерваної програми.

Мікроконтролер AT89C51 сімейства MCS-51 має систему переривань з п'ятьма джерелами запитів переривання. Кожне джерело має свій фіксований вектор переривання. Під вектором переривання, розуміється комірка пам'яті програм з фіксованою адресою, якій передається управління в разі приходу відповідного запиту переривання.

Адреси відповідних джерел переривань приведені в табл. 7.1

Таблиця 7.1

Джерела переривань	Адреси векторів переривань
Зовнішній вхід $\overline{INT0}$	0003H
Таймер TC0	000BH
Зовнішній вхід $\overline{INT1}$	0013H
Таймер TC1	001BH
Послідовний порт	0023H

Для прийому зовнішніх запитів на переривання служать лінії $\overline{INT0}$ та $\overline{INT1}$, які можуть бути запрограмовані на спрацювання як по фронту, так і по

рівню, залежно від значення бітів IT0 и IT1 регістра TCON. Бітову структуру регістру TCON приведено на рис.5.2.

Якщо значення біта IT0 (для виводу $\overline{INT0}$) або IT1 (для виводу $\overline{INT1}$) дорівнює 0, то запит на переривання виникає при нульовому рівні сигналу на даному вході (переривання по рівню). Якщо ж який-небудь з цих бітів дорівнює 1, то запит на переривання виникає при виявленні негативного перепаду рівня сигналу на цьому вході (переривання по фронту).

При виникненні запиту на переривання встановлюється прапор IE0 або IE1. Якщо переривання було викликане по фронту на зовнішньому виводі $\overline{INT0}$ або $\overline{INT1}$, то прапор запиту на переривання (IE0 або IE1) скидається апаратно при передачі управління обслуговуючій його підпрограмі обробки переривання.

Якщо запрограмоване переривання по рівню, то стан прапора запиту відповідає рівню сигналу на його зовнішньому виводі.

Для виявлення запиту необхідні рівні сигналів мають бути присутніми на вході на протязі, як мінімум, одного машинного циклу.

Прапор будь-якого переривання, крім того, може бути встановлений програмно для виклику відповідної підпрограми обробки.

Управління дозволами переривань.

Кожен вид переривання індивідуально дозволяється або забороняється установкою або очищенням відповідних біт SFR-регістру IE (Interrupt Enable). На рис.7.1 зображено бітову структуру цього регістру, а призначення окремих бітів описано в табл. 7.2.

7	6	5	4	3	2	1	0
EA	-	-	ES	ET1	EX1	ET0	EX0

Рис. 7.1. Бітова структура регістра IE

Таблиця 7.2

Біт	Позиція	Призначення
EA	IE.7	Біт загального дозволу переривань
ES	IE.4	Дозвіл переривань послідовного порту
ET1	IE.3	Дозвіл переривань таймера/лічильника TC1
EX1	IE.2	Дозвіл переривань по входу INT1
ET0	IE.1	Дозвіл переривань таймера/лічильника TC0
EX0	IE.0	Дозвіл переривань по входу INT0

Для дозволу переривання необхідно встановити відповідний біт дозволу в «1».

Біт EA визначає дозвіл/заборону роботи системи переривань. Якщо EA = 1, то переривання, які дозволені установкою індивідуальних засобів, дозволені в системі. Якщо він дорівнює EA = 0, то всі переривання в системі заборонені, навіть якщо окремі прапори дозволу конкретних переривань встановлені.

Управління пріоритетами переривань

При роботі мікроконтролера можлива ситуація, коли одночасно надходять запити на переривання від різних джерел. Для уникнення конфліктів в MCS51 реалізована дворівнева апаратно-програмна система пріоритетів, відповідно до якої пристрій управління вибирає джерело переривання, який може бути оброблений раніше. У межах кожного рівня пріоритетів встановлена наступна ієрархія пріоритетності обслуговування запитів: INT0 - TF0 - INT1 - TF1 - TI / RI.

Для будь-якого з п'яти джерел (TF1, TF0, INT1, INT0, TI або RI) програмним способом може бути встановлений один з двох рівнів пріоритету - "високий" або "низький". У разі збігу рівнів пріоритетів кількох одночасних запитів переривань, вибір вищого пріоритету пристрій управління проводить відповідно апаратної шкали INT0 - TF0 - INT1 - TF1 - TI / RI. Наприклад, якщо високий пріоритет обраний програмістом для переривань по входу INT0 і таймеру/лічильнику TC1, то в разі одночасного запиту від джерел IE0, TF1 спочатку

буде оброблятися запит INT0, а потім TC1 і тільки після цього - інші запити в порядку TF0-IE1-(TI / RI). Рівень пріоритету кожного джерела встановлюють за допомогою регістра пріоритетів переривань IP.

На рис.7.2 зображено бітову структуру цього регістру, а призначення окремих бітів описано в табл. 7.3.

7	6	5	4	3	2	1	0
-	-	-	PS	PT1	PX1	PT0	PX0

Рис. 7.2. Бітова структура регістра IP

Таблиця 7.3

Біт	Позиція	Призначення
PS	IP.4	Пріоритет переривань послідовного порту
PT1	IP.3	Пріоритет переривань таймера/лічильника TC1
PX1	IP.2	Пріоритет переривань по входу INT1
PT0	IP.1	Пріоритет переривань таймера/лічильника TC0
PX0	IP.0	Пріоритет переривань по входу INT0

Для встановлення високого рівня пріоритету переривання необхідно встановити відповідний біт в «1». Скидання біту керування приводить до призначення низького рівня пріоритету.

Приклад 7.1

Ініціалізувати систему переривань на наступний режим роботи:

- Дозволити переривання по фронту від входу INT0 з вищим рівнем пріоритету;
- Дозволити переривання з нижчим рівнем пріоритету по таймеру/лічильнику TC1, що працює в режимі таймера з програмним керуванням.

Сформувати відповідну область векторів переривань та порожні підпрограми переривань. Дозволити роботу системи переривань.

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Для вирішення цієї задачі необхідно сформулювати наступні константи ініціалізації:

- регістра TCON - для INT0 визначити роботу по фронту;
- регістра IE – визначити дозволи переривань по джерелам INT0, TC1;
- регістра IP - визначити пріоритети по джерелам INT0, TC1.

Необхідно дозволити режим альтернативних функцій по входу INT0 та перенести початок стеку з початкової адреси 7H, для звільнення місця під регістри змінних.

У табл.7.4 – 7.6 приведено відповідні налаштування регістрів TCON, IE, IP.

Таблиця 7.4 Регістр TCON

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
0	0	0	0	0	0	0	1

Таблиця 7.5 Регістр IE

7	6	5	4	3	2	1	0
EA	-	-	ES	ET1	EX1	ET0	EX0
0	0	0	0	1	0	0	1

Таблиця 7.6 Регістр IP

7	6	5	4	3	2	1	0
-	-	-	PS	PT1	PX1	PT0	PX0
0	0	0	0	0	0	0	1

ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ
Методичні вказівки до розрахунково-графічної роботи

Текст програми має наступний вигляд.

```
;-----  
;--- Опис констант та змінних  
INI_TCON equ 00000001b ;константа завантаження регістру TCON  
INI_IE equ 00001001b ;константа завантаження регістру IE  
INI_IP equ 00000001b ;константа завантаження регістру IP  
;--- Область векторів переривань  
ORG 0H ;адреса рестарту після пуску процесора  
SJMP INIT ;перехід на початок основної програми  
ORG 03H ;адреса рестарту після пуску процесора  
SJMP INT_INT0 ;перехід на програму переривань по INT0  
ORG 1BH ;адреса рестарту після пуску процесора  
SJMP INT_TC1 ;перехід на програму переривань по TC1  
;--- Програма  
ORG 20H ;початкова адреса основної програми  
INIT:  
MOV SP,#20h ;перенесення початку стеку  
  
MOV TCON,#INI_TCON ;ініціалізація регістру TCON  
MOV IE,#INI_IE ;ініціалізація регістру IE  
MOV IP,#INI_IP ;ініціалізація регістру IP  
  
SETB INT0 ;дозвіл альтернативних функцій по входу INT0  
SETB EA ;дозвіл роботи системи переривань  
MAIN:  
SJMP MAIN ;головний цикл програми  
;--- Підпрограма переривань по входу INT0  
INT_INT0:  
RETI  
;--- Підпрограма переривань по таймеру/лічильнику TC1  
INT_TC1:  
RETI  
  
END ;директива про закінчення програми  
;-----
```

ЛІТЕРАТУРА

1. Абрайтис, Б.Б. Микропроцессоры и микропроцессорные комплекты интегральных микросхем : справочник. В 2-х т. Т. 1 / Б.Б. Абрайтис и др. – М. : Радио и связь, 1988. – 368 с.
2. СверхБИС универсальных однокристальных ЭВМ /А. В. Кобылинский, Г. П. Липовецкий, Н. Г. Сабадаш и др. - К.: Техника, 1987, – 166с.
3. Сташин В. В., Урусов А. В., Мологонцева О. Ф. Проектирование цифровых устройств на однокристальных микроконтролерах: М.: Энергоатомиздат, 1990, – 224с.
4. Липовецкий Г. П. и др. Однокристальные микроЭВМ. Семейство МК48, Семейство МК51. Техническое описание и руководство по применению. - М.: МП "Бином", 1992. – 339 с.
5. Лебедев О. Н. Микросхемы памяти и их применение, М., Радио и связь, 1990. – 160 с.
6. Гутников В.С. Интегральная электроника в измерительных устройствах. - 2 изд, перераб и доп. - Л. Энергоатомиздат, ленинградское отд. , 1988. – 304 с.
7. Микроконтроллеры? Это же просто! Том 1. Фрунзе А.В. — М.: ООО «ИД СКИМЕН», 2002. — 336 с.
8. Фрунзе А.В. Микроконтроллеры? Это же просто! ; - М.; ООО «ИД СКИМЕН», 2002.
9. Микроконтроллеры? Это же просто! Том 2. Фрунзе А.В. — М.: ООО «ИД СКИМЕН», 2002. — 392 с.
10. Микроконтроллеры? Это же просто! Том 3. Фрунзе А. В., Фрунзе А. А. — М.: ООО «ИД СКИМЕН», 2003. — 224 с.
11. Микроконтроллеры? Это же просто! Том. 4 . Фрунзе А. В. — М.: Додэка-XXI, 2008. — 464 с.

12. Каспер Эрнст. Программирование языком ассемблера для микроконтроллеров семейства i8051. – М.: Горячая линия – Телеком, 2003.

13. Схемотехника электронных систем. Том 3. Микропроцессоры и микроконтроллеры / Бойко В. И., Гуржий А. М., Жуйков В. Я., Зорі А.А., Співак В.М., Терещенко Т. О, Петергеря Ю. С. - СПб.: БХВ Петербург, 2004. – 464 с.

14. Мікропроцесорна техніка. Друге видання. Доповнене./ Ю. І. Якименко, Т. О. Терещенко, Є. І. Сокол, В. Я. Жуйков, Ю. С. Петергеря. За ред. Т. О. Терещенко. – Київ, 2004. – 440 с .

15. <http://www.kaf-pe.ntu-kpi.kiev.ua/> Жуйков В.Я., Терещенко Т. О., Петергеря Ю. С. та ін. «Мікропроцесори і мікроконтролери» - Електронний підручник.